

# Bayesian Conditional Monte Carlo Algorithms for Sequential Single and Multi-Object filtering

Yohan Petetin, François Desbouvries

Received: date / Accepted: date

**Abstract** Bayesian filtering aims at tracking sequentially a hidden process from an observed one. In particular, sequential Monte Carlo (SMC) techniques propagate in time weighted trajectories which represent the posterior probability density function (pdf) of the hidden process given the available observations. On the other hand, Conditional Monte Carlo (CMC) is a variance reduction technique which replaces the estimator of a moment of interest by its conditional expectation given another variable. In this paper we show that up to some adaptations, one can make use of the time recursive nature of SMC algorithms in order to propose natural temporal CMC estimators of some point estimates of the hidden process, which outperform the associated crude Monte Carlo (MC) estimator whatever the number of samples. We next show that our Bayesian CMC estimators can be computed exactly, or approximated efficiently, in some hidden Markov chain (HMC) models; in some jump Markov state-space systems (JMSS); as well as in multitarget filtering. Finally our algorithms are validated via simulations.

**Keywords** Conditional Monte Carlo · Bayesian Filtering · Hidden Markov Models · Jump Markov state space systems · Rao-Blackwell Particle Filters · Probability Hypothesis Density.

## 1 Introduction

### 1.1 SMC algorithms for single- or multi-object Bayesian filtering

In single object Bayesian filtering we consider two random processes  $\{X_n\}_{n \geq 0}$  and  $\{Y_n\}_{n \geq 0}$  with given joint probability law.  $Y_i$  is observed, i.e. we have at our disposal realizations  $\mathbf{y}_{0:n} = \{y_i\}_{i=0}^n$  of  $\mathbf{Y}_{0:n} = \{Y_i\}_{i=0}^n$  (as far as notations

---

Telecom and Mines Institute / Telecom SudParis / CITI Department and CNRS UMR 5157, 9 rue Charles Fourier, 91011 Evry, France. Email: {yohan.petetin,francois.desbouvries}@telecom-sudparis.eu

are concerned, upper case letters denote random variables (r.v.), lower case ones their realizations, and bold letters vectors;  $p(x)$ , say, denotes the pdf of r.v.  $X$  and  $p(x|y)$ , say, the conditional pdf of  $X$  given  $Y = y$ ; if  $i \leq j$   $p_{i:j|n}$  is a shorthand notation for  $p(\mathbf{x}_{i:j}|\mathbf{y}_{0:n})$ ; if  $x^i$  are samples from  $p(x)$  then the set  $\{x^i\}_{i=1}^N$  can also be denoted  $\mathbf{x}^{1:N}$ ; subscripts are reserved for times indices and superscripts for realizations). Process  $\{X_n\}$  is hidden, and our aim is to compute, for each time instant  $n$ , some moment of interest

$$\Theta_n = \int f(\mathbf{x}_{0:n})p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n})d\mathbf{x}_{0:n} \quad (1)$$

of the a posteriori pdf  $p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n})$  of  $\mathbf{X}_{0:n}$  given  $\mathbf{y}_{0:n}$ . Unfortunately, in most models (1) cannot be computed exactly. Suboptimal solutions for computing  $\Theta_n$  include SMC techniques [1] [2], which propagate over time weighted trajectories  $\{\mathbf{x}_{0:n}^i, w_n^i\}_{i=1}^N$  with  $\sum_{i=1}^N w_n^i = 1$ . In other words,  $\hat{p}_{0:n|n} = \sum_{i=1}^N w_n^i \delta_{\mathbf{x}_{0:n}^i}$ , in which  $\delta$  is the Dirac mass, is a discrete (and random) approximation of  $p(\mathbf{x}_{0:n}|\mathbf{y}_{0:n})$ .

On the other hand, *multi-object* filtering (see e.g. [3]) essentially reduces to computing  $\Theta_n = \int f(\mathbf{x}_n)v_n(\mathbf{x}_n)d\mathbf{x}_n$  in which  $v_n(\mathbf{x}_n)$  is now the so-called Probability Hypothesis Density (PHD), i.e. the a posteriori spatial density of the expected number of targets, given all measurements (be they due to detected targets or to false alarms). Again, SMC techniques propagate an approximation of  $v_n$  with a set of weighted samples  $\{\mathbf{x}_n^i, w_n^i\}_{i=1}^N$ ; here  $\sum_{i=1}^N w_n^i$ , which in general is different from 1, is an estimator of the number of targets.

Now, SMC algorithms, be they for single- or multi-object Bayesian filtering, usually focus on how to propagate approximations  $\hat{p}_{0:n|n}$  (or  $\hat{v}_n$ ) of  $p_{0:n|n}$  (or  $v_n$ ); once  $\hat{p}_{0:n|n}$  or  $\hat{v}_n$  has been computed,  $\Theta_n$  is finally estimated either as  $\hat{\Theta}_n = \sum_{i=1}^N w_n^i f(\mathbf{x}_{0:n}^i)$  or  $\sum_{i=1}^N w_n^i f(\mathbf{x}_n^i)$ . By contrast, in this paper we directly focus on  $\hat{\Theta}_n$  itself, and see under which conditions one can improve this point estimator at a reasonable computational cost.

## 1.2 Variance reduction via conditioning: Rao-Blackwellized particle filters (RB-PF)

This problem leads us to variance reduction techniques which form an important part of computer simulation (see e.g. [4]). Among them, methods based on conditioning variables rely on the following well known result. Let  $X_1$  and  $X_2$  be two r.v. and  $f$  some function. Then

$$\mathbb{E}(\mathbb{E}(f(X_2)|X_1)) = \mathbb{E}(f(X_2)), \quad (2)$$

$$\text{var}(\mathbb{E}(f(X_2)|X_1)) = \text{var}(f(X_2)) - \mathbb{E}(\text{var}(f(X_2)|X_1)). \quad (3)$$

So if the aim is to compute  $\Theta = \mathbb{E}(f(X_2))$  and we have at our disposal  $\{X_1^i\}_{i=1}^N \stackrel{\text{i.i.d.}}{\sim} p(x_1)$ ,  $\{X_2^i\}_{i=1}^N \stackrel{\text{i.i.d.}}{\sim} p(x_2)$  then the so-called CMC estimator  $\hat{\Theta} = \frac{1}{N} \sum_{i=1}^N \mathbb{E}(f(X_2)|X_1^i)$  has lower variance than the corresponding crude

MC one  $\hat{\Theta} = \frac{1}{N} \sum_{i=1}^N f(X_2^i)$ . Of course, the interest of  $\tilde{\Theta}$  vs.  $\hat{\Theta}$  depends on the choice of  $X_1$ : ideally, one should easily sample from  $p(x_1)$ ; the variance reduction in (3) should be as large as possible; but in the meantime function  $g(x_1) = E(f(X_2)|x_1)$  should remain computable at a reasonable computational cost.

Variance reduction techniques based on CMC methods have been adapted to Bayesian filtering; in this context, these methods are either known as marginalized or RB-PF [5] [6] [7] [8]. The rationale is as follow. Let now  $\Theta_n$  in (1) be rewritten as  $\Theta = E(f(X_1, X_2))$ . It is usually not possible to sample from  $p(x_1, x_2)$ , and often  $p(x_1, x_2) \propto p'(x_1, x_2)$  is only known up to a constant, whence the use of Bayesian (or normalized) importance sampling (IS) techniques [9]. So let

$$\hat{\Theta}(\mathbf{x}_1^{1:N}, \mathbf{x}_2^{1:N}) = \sum_{i=1}^N w_2^i(\mathbf{x}_1^{1:N}, \mathbf{x}_2^{1:N}) f(x_1^i, x_2^i) \text{ with } (x_1^i, x_2^i) \sim q_2, \quad (4)$$

$$\hat{\Theta}^{\text{RB}}(\mathbf{x}_1^{1:N}) = \sum_{i=1}^N w_1^i(\mathbf{x}_1^{1:N}) E(f(x_1^i, X_2)|x_1^i) \text{ with } x_1^i \sim q_1, \quad (5)$$

with  $\sum_{i=1}^N w_1^i = \sum_{i=1}^N w_2^i = 1$ . Estimator  $\hat{\Theta}^{\text{RB}}$  depends on samples  $\{x_1^i\}_{i=1}^N$  only and is known as the RB estimator of  $\Theta$ . However  $\hat{\Theta}^{\text{RB}}$  is known to outperform  $\hat{\Theta}$  only under specific assumptions on  $q_1$ ,  $q_2$ ,  $\mathbf{w}_1^{1:N}$  and  $\mathbf{w}_2^{1:N}$ . In particular, if  $w_1^i \propto w_1^{u,i} = p'(x_1^i)/q_1(x_1^i)$ ,  $w_2^i \propto w_2^{u,i} = p'(x_1^i, x_2^i)/q_2(x_1^i, x_2^i)$  and  $q_1(x_1) = \int q_2(x_1, x_2) dx_2$ , then the variance of  $w_1^{u,i}$  can only lower than that of  $w_2^{u,i}$  [6]. If moreover  $(x_1^i, x_2^i)$  are independent, an asymptotic analysis based on (2) and (3) proves that  $\hat{\Theta}^{\text{RB}}$  indeed outperforms  $\hat{\Theta}$  [7]. However, independence never holds in the presence of resampling; in the general case, the comparison of both estimators depends on the choice of the importance distributions  $q_1$  and  $q_2$ , and can be proved (asymptotically) only under specific sufficient conditions [10] [11].

RB-PF have been applied in the specific case where the state vectors  $\mathbf{x}_{0:n}$  can be partitioned into a “linear” component  $\mathbf{x}_2 = \mathbf{x}_{0:n}^l$  and a “non-linear” one  $\mathbf{x}_1 = \mathbf{x}_{0:n}^{nl}$ . Models in which computing  $\hat{\Theta}^{\text{RB}}$  is possible include linear and Gaussian JMSS [7] [5] or partially linear and Gaussian HMC [8]. In other models, it may be possible to approximate  $\hat{\Theta}^{\text{RB}}$  by using numerical approximations of  $w_1(x)$  and of  $E(f(X_1, X_2)|x_1)$ . However, due to the spatial structure of the decomposition of  $\mathbf{x}_{0:n}$ , approximating  $\Theta_n$  in (1) involves propagating numerical approximations over time.

### 1.3 Bayesian CMC estimators

#### 1.3.1 Spatial vs. temporal RB-PF

In this paper we propose another class of RB-PF; the main difference is that our partitioning  $(X_1, X_2)$  of  $\mathbf{x}_{0:n}$  is now temporal rather than spatial. The

question arises naturally in the Bayesian filtering context: at time  $n$  we usually build  $\Theta_n$  from  $\hat{p}_{0:n|n}$ , but indeed  $\hat{p}_{0:n-1|n-1}$  was also available for free since, by nature, *sequential* MC algorithms construct  $\hat{p}_{0:n|n}$  from  $\hat{p}_{0:n-1|n-1}$ . Now, comparing with spatially partitioned RB-PF, a temporal partition of  $\mathbf{x}_{0:n}$  has a number of statistical and computational structural consequences, as we now see. So let again

$$\Theta = \int f(x_1, x_2) p(x_1, x_2) dx_1 dx_2 \quad (6)$$

$$= \int \left[ \int f(x_1, x_2) p(x_2|x_1) dx_2 \right] p(x_1) dx_1. \quad (7)$$

Let us start from the following approximation of  $p(x_1)$ :

$$p(x_1) \approx \hat{p}(x_1) = \sum_{i=1}^N w^i(\mathbf{x}_1^{1:N}) \delta_{x_1^i}. \quad (8)$$

For  $1 \leq i \leq N$  let next  $x_2^i \sim p(x_2|x_1^i)$ . This yields the following approximation of  $p(x_1, x_2)$ :

$$\hat{p}(x_1, x_2) = \sum_{i=1}^N w^i(\mathbf{x}_1^{1:N}) \delta_{(x_1^i, x_2^i)}; \quad (9)$$

note that each weight  $w^i$  may depend on  $\{x_1^i\}_{i=1}^N$ , but not on  $\{x_2^i\}_{i=1}^N$ . The reason why is that we now use a temporal partition, and not a spatial one: in the spatial subdivision case,  $p(x_2|x_1)$  would reduce to  $p(\mathbf{x}_{0:n}^l | \mathbf{x}_{0:n}^{nl}, \mathbf{y}_{0:n})$ , which means that we would need to sample at each time step the whole set  $\{\mathbf{x}_{0:n}^{l,i}\}_{i=1}^N$ , instead of simply extending the trajectories.

Finally we have two options: computing the full expectation in (6) by using (9), or only the outer one in (7) by using (8). So let

$$\hat{\Theta}(\mathbf{x}_1^{1:N}, \mathbf{x}_2^{1:N}) = \sum_{i=1}^N w^i(\mathbf{x}_1^{1:N}) f(x_1^i, x_2^i), \quad (10)$$

$$\tilde{\Theta}(\mathbf{x}_1^{1:N}) = \sum_{i=1}^N w^i(\mathbf{x}_1^{1:N}) \left[ \int f(x_1^i, x_2) p(x_2|x_1^i) dx_2 \right]. \quad (11)$$

In this paper, we shall call  $\hat{\Theta}(\mathbf{x}_1^{1:N}, \mathbf{x}_2^{1:N})$  (resp.  $\tilde{\Theta}(\mathbf{x}_1^{1:N})$ ) the Bayesian crude MC (resp. Bayesian CMC) estimator of  $\Theta$ .

### 1.3.2 Discussion

Let us now compare  $\tilde{\Theta}$  to  $\hat{\Theta}$ . As in section 1.2,  $\tilde{\Theta}$  outperforms  $\hat{\Theta}$  for all  $N$ , but not for the same reasons. Indeed we have

$$\mathbb{E}(w^i(\mathbf{x}_1^{1:N}) f(x_1^i, x_2) | \mathbf{x}_1^{1:N}) = w^i(\mathbf{x}_1^{1:N}) \int f(x_1^i, x_2) p(x_2|x_1^i) dx_2. \quad (12)$$

So from (3), the variance of each term of (11) is lower than or equal to that of the corresponding term in (10); however this is not sufficient to conclude that  $\text{var}(\tilde{\Theta}) \leq \text{var}(\hat{\Theta})$  since the terms may be dependent. Fortunately (12) implies that  $\tilde{\Theta} = E(\hat{\Theta}|\mathbf{x}_1^{1:N})$ , so  $\tilde{\Theta}$  is preferable to  $\hat{\Theta}$ , due to (2) and (3).

Let us now turn to practical considerations. Of course,  $\tilde{\Theta}$  is of interest only if the conditional expectation in (11) can be computed easily. In the rest of this paper we will see that this indeed is the case in some Markovian models and for other models, we will propose and discuss some approximations which make the Bayesian CMC estimator a tool of practical interest for practitioners which may be used as an alternative to purely Monte Carlo classical PF. From a modeling point of view, by contrast with spatially partitioned RB-PF, the state space no longer needs to be multi-dimensional; here a key point is the availability (and integrability) of  $p(x_2|x_1)$ , which, in the temporal partitions considered below, will coincide with the so-called optimal conditional importance distribution. From a numerical point of view, another interesting feature of sequential RB-PF is that numerical approximations, when necessary, do not need to be propagated over time.

Let us finally address complexity. As we shall see, in some cases  $\tilde{\Theta}$  can even be computed under the same assumptions and for the same computational cost as  $\hat{\Theta}$  (see sections 2.2.1 and 3.2.2). Also one should note that in the partition  $(X_1, X_2)$  of a given set of variables  $(\mathbf{X}_{0:n}, \text{ say})$   $X_1$  should be as small as possible. More precisely, let  $\Theta = E(f(X_1, X_2, X_3))$  and let  $\hat{p}(x_1, x_2)$  be available. Then two Bayesian CMC estimators can be thought of :  $\tilde{\Theta}^{X_3}$  built from  $\Theta = E[E(f(X_1, X_2, X_3)|X_1, X_2)]$ , in which the inner expectation (w.r.t.  $X_3$ ) is computed exactly, and  $\tilde{\Theta}^{(X_2, X_3)}$  built from  $\Theta = E[E(f(X_1, X_2, X_3)|X_1)]$  and from  $\hat{p}(x_1)$ . Estimator  $\tilde{\Theta}^{(X_2, X_3)}$  is preferable to  $\tilde{\Theta}^{X_3}$ , but computing  $\tilde{\Theta}^{(X_2, X_3)}$  requires an additional exact expectation computation, since  $E(f(X_1, X_2, X_3)|X_1) = E[E(f(X_1, X_2, X_3)|X_1, X_2)]$ . As we shall see in section 3.2.2, in some Markovian models both estimators can indeed be computed; and computing  $\tilde{\Theta}^{(X_2, X_3)}$  only requires an additional computational cost.

The rest of this paper is organized as follows. First in section 2 we see that in some HMC models (including the Autoregressive Conditional Heteroscedasticity (ARCH) ones), a Bayesian CMC estimator  $\tilde{\Theta}_n$  can replace the classical one  $\hat{\Theta}_n$  in the case where the sampling importance resampling (SIR) algorithm with optimal importance distribution is used. In Section 3 we develop our Bayesian CMC estimators for JMSS; in section 3.1 we address the linear and Gaussian case, where our solution can be seen as a further (temporal) RB step of an already (spatial) RB-PF algorithm; in section 3.2 we develop Bayesian CMC estimators for general JMSS. Finally in Section 4 we address a multi-target scenario and adapt Bayesian CMC to the PHD filter. In all these sections we propose relevant approximate estimators when the Bayesian CMC estimator cannot be computed exactly, and we validate our algorithms via simulations. We finally end the paper with a Conclusion.

## 2 Bayesian CMC PF for some HMC models

### 2.1 Deriving a Bayesian CMC estimator $\tilde{\Theta}_n$

Let  $\{\mathbf{X}_n\}_{n \geq 0}$  (resp.  $\{\mathbf{Y}_n\}_{n \geq 0}$ ) be a  $p$ - (resp.  $q$ -) dimensional state vector (resp. observation). We assume that  $(\mathbf{X}_n, \mathbf{Y}_n)$  follows the well known HMC model:

$$p(\mathbf{x}_{0:n}, \mathbf{y}_{0:n}) = p(\mathbf{x}_0) \prod_{i=1}^n f_{i|i-1}(\mathbf{x}_i | \mathbf{x}_{i-1}) \prod_{i=0}^n g_i(\mathbf{y}_i | \mathbf{x}_i), \quad (13)$$

in which  $f_{i|i-1}(\mathbf{x}_i | \mathbf{x}_{i-1})$  is the transition pdf of Markov chain  $\{\mathbf{X}_n\}_{n \geq 0}$  and  $g_i(\mathbf{y}_i | \mathbf{x}_i)$  the likelihood of  $\mathbf{y}_i$  given  $\mathbf{x}_i$ . The Bayesian filtering problem consists in computing some moment of interest  $\Theta_n = E_{p_n}(f(\mathbf{X}_n))$ , which we rewrite as

$$\Theta_n = \int f(\mathbf{x}_n) p(\mathbf{x}_{0:n-1}, \mathbf{x}_n | \mathbf{y}_{0:n}) d\mathbf{x}_{0:n-1} d\mathbf{x}_n. \quad (14)$$

So (14) coincides with (6), with  $X_1 = \mathbf{X}_{0:n-1}$ ,  $X_2 = \mathbf{X}_n$ ,  $f(x_1, x_2)$  depends on  $x_2$  only, and  $p(x_1, x_2)$  is the a posteriori (i.e., given  $\mathbf{y}_{0:n}$ ) joint pdf

$$p(\mathbf{x}_{0:n-1}, \mathbf{x}_n | \mathbf{y}_{0:n}) = \underbrace{p(\mathbf{x}_{0:n-1} | \mathbf{y}_{0:n})}_{p(x_1)} \underbrace{p(\mathbf{x}_n | \mathbf{x}_{n-1}, \mathbf{y}_n)}_{p(x_2 | x_1)}. \quad (15)$$

According to (8) we first need an approximation of  $p(x_1)$ , which in model (13) reads:

$$p(\mathbf{x}_{0:n-1} | \mathbf{y}_{0:n}) = \frac{p(\mathbf{y}_n | \mathbf{x}_{n-1}) p(\mathbf{x}_{0:n-1} | \mathbf{y}_{0:n-1})}{\int p(\mathbf{y}_n | \mathbf{x}_{n-1}) p(\mathbf{x}_{0:n-1} | \mathbf{y}_{0:n-1}) d\mathbf{x}_{0:n-1}}, \quad (16)$$

in which  $p(\mathbf{y}_n | \mathbf{x}_{n-1}) = \int g_n(\mathbf{y}_n | \mathbf{x}_n) f_{n|n-1}(\mathbf{x}_n | \mathbf{x}_{n-1}) d\mathbf{x}_n$ . On the other hand, PF algorithm propagate approximations of  $p_{0:n-1|n-1}$  or of  $p_{n-1|n-1}$ . So let us start from  $\hat{p}(\mathbf{x}_{0:n-1} | \mathbf{y}_{0:n-1}) = \sum_{i=1}^N w_{n-1}^i \delta_{\mathbf{x}_{0:n-1}^i}$ . According to Rubin's SIR mechanism [12] [13] [14]  $\hat{p}(\mathbf{x}_{0:n-1} | \mathbf{y}_{0:n}) = \sum_{i=1}^N \tilde{w}_{n-1}^i \delta_{\mathbf{x}_{0:n-1}^i}$ , where  $\tilde{w}_{n-1}^i \propto w_{n-1}^i p(\mathbf{y}_n | \mathbf{x}_{n-1})$ , is an approximation of  $p(\mathbf{x}_{0:n-1} | \mathbf{y}_{0:n})$ . Next  $p(x_2 | x_1)$  in (15) coincides with the so-called optimal conditional importance pdf, i.e. the importance density  $p(\mathbf{x}_n | \mathbf{x}_{n-1}, \mathbf{y}_n) \propto g_n(\mathbf{y}_n | \mathbf{x}_n) f_{n|n}(\mathbf{x}_n | \mathbf{x}_{n-1})$  which minimizes the conditional variance of weights  $w_n^i$ , given past trajectories and observations [15] [16] [17] and [6]. This leads to the so-called SIR algorithm with optimal importance distribution and optional resampling step:

**SIR algorithm.** Let  $\hat{p}_{0:n-1|n-1} = \sum_{i=1}^N w_{n-1}^i \delta_{\mathbf{x}_{0:n-1}^i}$  be an MC approximation of  $p_{0:n-1|n-1}$ .

1. For all  $i$ ,  $1 \leq i \leq N$ , sample  $\tilde{\mathbf{x}}_n^i \sim p(\mathbf{x}_n | \mathbf{x}_{n-1}^i, \mathbf{y}_n)$ ;
2. For all  $i$ ,  $1 \leq i \leq N$ , set  $\tilde{w}_n^i \propto w_{n-1}^i p(\mathbf{y}_n | \mathbf{x}_{n-1}^i)$ ,  $\sum_{i=1}^N \tilde{w}_n^i = 1$ ;
3. (Optional). For all  $i$ ,  $1 \leq i \leq N$ , (re)sample  $\mathbf{x}_{0:n}^i \sim \sum_{i=1}^N \tilde{w}_n^i \delta_{[\mathbf{x}_{0:n-1}^i, \tilde{\mathbf{x}}_n^i]}$ , and set  $w_n^i = \frac{1}{N}$ ; otherwise set  $(\mathbf{x}_n^i, w_n^i) = (\tilde{\mathbf{x}}_n^i, \tilde{w}_n^i)$ .

This third resampling step is usually performed only if some criterion holds, and aims at preventing weights degeneracy, see e.g. [1], [2]. Then

$$\hat{p}_{0:n|n}^{\text{SIR}} = \sum_{i=1}^N \tilde{w}_n^i \delta_{\mathbf{x}_{0:n-1}^i, \tilde{\mathbf{x}}_n^i} \quad (17)$$

is a (SIR-based) SMC approximation of  $p_{0:n|n}$ , and  $\hat{p}_{0:n-1,n|n}^{\text{SIR}}$  plays the role of  $\hat{p}(x_1, x_2)$  in (9). Finally from (10) and (11), the SIR-based crude and CMC estimators of moment  $\Theta_n$  defined in (14) are respectively

$$\hat{\Theta}_n^{\text{SIR}}(\mathbf{x}_{0:n-1}^{1:N}, \tilde{\mathbf{x}}_n^{1:N}) = \sum_{i=1}^N \tilde{w}_n^i(\mathbf{x}_{0:n-1}^{1:N}) f(\tilde{\mathbf{x}}_n^i), \quad (18)$$

$$\tilde{\Theta}_n^{\text{SIR}}(\mathbf{x}_{0:n-1}^{1:N}) = \sum_{i=1}^N \tilde{w}_n^i(\mathbf{x}_{0:n-1}^{1:N}) \int f(\mathbf{x}_n) p(\mathbf{x}_n | \mathbf{x}_{n-1}^i, \mathbf{y}_n) d\mathbf{x}_n. \quad (19)$$

## 2.2 Computing $\tilde{\Theta}_n^{\text{SIR}}$ in practice

### 2.2.1 Exact computation

From (12) we know that  $\tilde{\Theta}_n^{\text{SIR}}$  outperforms  $\hat{\Theta}_n^{\text{SIR}}$ ; but  $\tilde{\Theta}_n^{\text{SIR}}$  can be used only if  $\tilde{w}_n^i$  and integral  $\int f(\mathbf{x}_n) p(\mathbf{x}_n | \mathbf{x}_{n-1}^i, \mathbf{y}_n) d\mathbf{x}_n$  can be computed. As we now see, this is the case in some particular HMC models and for some functions  $f(\cdot)$ . Let us e.g. consider the semi-linear stochastic models with additive Gaussian noise, given by

$$\mathbf{X}_n = \mathbf{f}_n(\mathbf{X}_{n-1}) + \mathbf{K}_n(\mathbf{X}_{n-1}) \times \mathbf{U}_n, \quad (20)$$

$$\mathbf{Y}_n = \mathbf{H}_n \mathbf{X}_n + \mathbf{V}_n, \quad (21)$$

in which  $\{\mathbf{U}_n\}$  and  $\{\mathbf{V}_n\}$  are i.i.d., mutually independent and independent of  $\mathbf{X}_0$ ,  $\mathbf{U}_n \sim \mathcal{N}(0; \mathbf{I})$  and  $\mathbf{V}_n \sim \mathcal{N}(0; \mathbf{R}_n^v)$ . The one-dimensional ARCH model is one such model with  $f_n(x_{n-1}) = 0$ ,  $k_n(x_{n-1}) = \sqrt{\beta_0 + \beta_1 x_{n-1}^2}$  and  $H_n = 1$ . In model (20) (21)  $p(\mathbf{x}_n | \mathbf{x}_{n-1}, \mathbf{y}_n)$  and  $p(\mathbf{y}_n | \mathbf{x}_{n-1})$  are Gaussian. More precisely, let  $\mathbf{Q}_n(\mathbf{x}_{n-1}) = \mathbf{K}_n(\mathbf{x}_{n-1}) \mathbf{K}_n(\mathbf{x}_{n-1})^T$ ; then

$$\mathbf{L}_n(\mathbf{x}_{n-1}) = \mathbf{H}_n \mathbf{Q}_n(\mathbf{x}_{n-1}) \mathbf{H}_n^T + \mathbf{R}_n^v, \quad (22)$$

$$\mathbf{m}_n(\mathbf{x}_{n-1}, \mathbf{y}_n) = \mathbf{f}_n(\mathbf{x}_{n-1}) + \mathbf{Q}_n(\mathbf{x}_{n-1}) \mathbf{H}_n^T \mathbf{L}_n^{-1}(\mathbf{x}_{n-1}) (\mathbf{y}_n - \mathbf{H}_n \mathbf{f}_n(\mathbf{x}_{n-1})), \quad (23)$$

$$\mathbf{P}_n(\mathbf{x}_{n-1}) = \mathbf{Q}_n(\mathbf{x}_{n-1}) - \mathbf{Q}_n(\mathbf{x}_{n-1}) \mathbf{H}_n^T \mathbf{L}_n^{-1}(\mathbf{x}_{n-1}) \mathbf{H}_n \mathbf{Q}_n(\mathbf{x}_{n-1}), \quad (24)$$

$$p(\mathbf{x}_n | \mathbf{x}_{n-1}, \mathbf{y}_n) = \mathcal{N}(\mathbf{x}_n, \mathbf{m}_n(\mathbf{x}_{n-1}, \mathbf{y}_n), \mathbf{P}_n(\mathbf{x}_{n-1})), \quad (25)$$

$$p(\mathbf{y}_n | \mathbf{x}_{n-1}) = \mathcal{N}(\mathbf{y}_n, \mathbf{H}_n \mathbf{f}_n(\mathbf{x}_{n-1}), \mathbf{L}_n(\mathbf{x}_{n-1})). \quad (26)$$

Finally in such models the Bayesian CMC estimator  $\tilde{\Theta}_n^{\text{SIR}}$  is workable for some functions  $f(\cdot)$ . If  $f(\mathbf{x})$  is a polynomial in  $\mathbf{x}$ , the problem reduces to computing

the first moments of the available Gaussian pdf (25). In the important particular case where  $f(\mathbf{x}) = \mathbf{x}$  (used to give an estimator of the hidden state), no further computation is indeed necessary; in this case the integral in (19) is equal to  $\mathbf{m}_n(\mathbf{x}_{n-1}^i, \mathbf{y}_n)$ .

*Remark 1* In this class of models, computing  $\tilde{\Theta}_n^{\text{SIR}}$  or  $\hat{\Theta}_n^{\text{SIR}}$  requires the same computational cost if  $f(\mathbf{x}) = \mathbf{x}$ . Both estimators indeed compute the parameters  $\mathbf{m}_n(\mathbf{x}_{n-1}^i, \mathbf{y}_n)$  and  $\mathbf{P}_n(\mathbf{x}_{n-1}^i)$  of  $p(\mathbf{x}_n|\mathbf{x}_{n-1}^i, \mathbf{y}_n)$ , and use these pdfs to sample the new particles  $\tilde{\mathbf{x}}_n^i$ , which in both cases are needed for the next time step. The only difference is that  $\hat{\Theta}_n^{\text{SIR}} = \sum_{i=1}^N \tilde{w}_n^i \tilde{\mathbf{x}}_n^i$ , while  $\tilde{\Theta}_n^{\text{SIR}} = \sum_{i=1}^N \tilde{w}_n^i \mathbf{m}_n(\mathbf{x}_{n-1}^i, \mathbf{y}_n)$ .

### 2.2.2 Approximate computation

Let us now discuss cases where the Bayesian CMC estimator  $\tilde{\Theta}_n^{\text{SIR}}$  cannot be computed exactly because  $p(\mathbf{y}_n|\mathbf{x}_{n-1})$  and/or moments of  $p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n)$  are not computable. Two approximations are proposed:

- Available techniques such as local linearizations [6], Taylor series expansion [18] or the Unscented Transformation (UT) [19] have already been proposed for approximating  $p(\mathbf{y}_n|\mathbf{x}_{n-1})$  and a moment of  $p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{y}_n)$ , so one can use any of them in (19). The resulting algorithm can be seen as an alternative to solutions like the Extended Kalman Filter (EKF) or the Unscented Kalman filter (UKF), where we look for approximating the filtering pdf  $p_{n|n}$  by a Gaussian and which rely on linearizations or the UT; or to SMC methods, where we look for a discrete approximation of  $p_{n|n}$ . In our approximate Bayesian CMC technique, we start from a discrete approximation of  $p_{n-1|n-1}$  produced by an SMC method, then similarly to the EKF/UKF, we look for a numerical approximation of  $\tilde{\Theta}_n$ , given that discrete approximation of  $p_{n-1|n-1}$ . However, deriving a good approximation of  $p(\mathbf{y}_n|\mathbf{x}_{n-1})$  can be an intricate issue, so we next look for approximations which do not rely on an approximation of  $p(\mathbf{y}_n|\mathbf{x}_{n-1})$ .
- In the SIR algorithm used so far,  $\tilde{\mathbf{x}}_n^i$  is drawn from  $p(\mathbf{x}_n|\mathbf{x}_{n-1}^i, \mathbf{y}_n)$ , whence a weight update factor equal to  $p(\mathbf{y}_n|\mathbf{x}_{n-1}^i)$ . On the other hand, sampling  $\tilde{\mathbf{x}}_n^i$  from an alternate (i.e., not necessarily optimal) pdf  $q(\mathbf{x}_n|\mathbf{x}_{n-1}^i)$  yields an approximation of  $p_{0:n-1|n}$  given by  $\hat{p}_{0:n-1|n} = \sum \tilde{w}_n^i \delta_{\mathbf{x}_{0:n-1}^i}$ , where weights  $\tilde{w}_n^i$  are now proportional to  $w_{n-1}^i f_{n|n-1}(\tilde{\mathbf{x}}_n^i|\mathbf{x}_{n-1}^i) g_n(\mathbf{y}_n|\tilde{\mathbf{x}}_n^i) / q(\tilde{\mathbf{x}}_n^i|\mathbf{x}_{n-1}^i)$ , and so depend also on the new samples  $\{\tilde{\mathbf{x}}_n^i\}_{i=1}^N$ . In that case, the associated Bayesian CMC and crude estimators become

$$\hat{\Theta}_n(\mathbf{x}_{0:n-1}^{1:N}, \tilde{\mathbf{x}}_n^{1:N}) = \sum_{i=1}^N \tilde{w}_n^i(\mathbf{x}_{0:n-1}^{1:N}, \tilde{\mathbf{x}}_n^{1:N}) f(\tilde{\mathbf{x}}_n^i), \quad (27)$$

$$\tilde{\Theta}_n(\mathbf{x}_{0:n-1}^{1:N}, \tilde{\mathbf{x}}_n^{1:N}) = \sum_{i=1}^N \tilde{w}_n^i(\mathbf{x}_{0:n-1}^{1:N}, \tilde{\mathbf{x}}_n^{1:N}) \int f(\mathbf{x}_n) p(\mathbf{x}_n|\mathbf{x}_{n-1}^i, \mathbf{y}_n) d\mathbf{x}_n, \quad (28)$$



which can no longer be compared easily (it was the case in section 1.3, because the weights  $w^i$  in  $\hat{\Theta}$  in (10) and  $\tilde{\Theta}$  in (11) depend on  $\{x_1^i\}_{i=1}^N$  only). On the other hand, the computation of (28) does not require that of  $p(\mathbf{y}_n|\mathbf{x}_{n-1})$ , but only that of  $\int f(\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{x}_{n-1}^i, \mathbf{y}_n)d\mathbf{x}_n$ . This is of interest in some models where approximating  $p(\mathbf{y}_n|\mathbf{x}_{n-1})$  may be challenging because of the form of  $g_n$ , while the first order moments of  $p(x_n|x_{n-1}, y_n)$  can be computed or approximated easily [18].

The two approximate implementations of the Bayesian CMC estimator which we just discussed will be compared via simulations in section §2.4.3.

## 2.3 Alternate Bayesian CMC solutions

### 2.3.1 A Bayesian CMC estimator based on the fully-adapted auxiliary particle filter (FA)

The SIR algorithm of section 2.1 is not the only SMC algorithm which enables to compute an approximation of  $p(\mathbf{x}_{0:n-1}, \mathbf{x}_n|\mathbf{y}_{0:n})$  in which weights depend on  $\{\mathbf{x}_{0:n-1}^i\}_{i=1}^N$  only. Starting from  $\hat{p}(\mathbf{x}_{0:n-1}|\mathbf{y}_{0:n-1}) = \sum_{i=1}^N w_{n-1}^i \delta_{\mathbf{x}_{0:n-1}^i}$ , the so-called FA algorithm [20] [21] is one such alternative:

**FA algorithm.** Let  $\hat{p}_{0:n-1|n-1} = \sum_{i=1}^N w_{n-1}^i \delta_{\mathbf{x}_{0:n-1}^i}$  be an MC approximation of  $p_{0:n-1|n-1}$ .

1. For all  $i$ ,  $1 \leq i \leq N$ , set  $\tilde{w}_n^i \propto w_{n-1}^i p(\mathbf{y}_n|\mathbf{x}_{0:n-1}^i)$ ,  $\sum_{i=1}^N \tilde{w}_n^i = 1$ ;
2. For all  $i$ ,  $1 \leq i \leq N$ , sample  $\tilde{\mathbf{x}}_{0:n-1}^i \sim \sum_{i=1}^N \tilde{w}_n^i \delta_{\mathbf{x}_{0:n-1}^i}$ ,
3. For all  $i$ ,  $1 \leq i \leq N$ , sample  $\mathbf{x}_n^i \sim p(\mathbf{x}_n|\tilde{\mathbf{x}}_{n-1}^i, \mathbf{y}_n)$  and set  $w_n^i = \frac{1}{N}$ ,  $\mathbf{x}_{0:n}^i = [\tilde{\mathbf{x}}_{0:n-1}^i, \mathbf{x}_n^i]$ .

Finally

$$\hat{p}_{0:n-1,n|n}^{\text{FA}} = \sum_{i=1}^N \frac{1}{N} \delta_{\tilde{\mathbf{x}}_{0:n-1}^i, \mathbf{x}_n^i} \quad (29)$$

is the FA-based SMC approximation of  $p_{0:n-1,n|n}$ , and the FA-based crude and CMC estimators of  $\Theta_n$  become respectively

$$\hat{\Theta}_n^{\text{FA}}(\tilde{\mathbf{x}}_{0:n-1}^{1:N}, \mathbf{x}_n^{1:N}) = \hat{\Theta}_n^{\text{FA}}(\mathbf{x}_n^{1:N}) = \frac{1}{N} \sum_{i=1}^N f(\mathbf{x}_n^i), \quad (30)$$

$$\tilde{\Theta}_n^{\text{FA}}(\tilde{\mathbf{x}}_{0:n-1}^{1:N}) = \tilde{\Theta}_n^{\text{FA}}(\tilde{\mathbf{x}}_{n-1}^{1:N}) = \frac{1}{N} \sum_{i=1}^N \int f(\mathbf{x}_n) p(\mathbf{x}_n|\tilde{\mathbf{x}}_{n-1}^i, \mathbf{y}_n) d\mathbf{x}_n. \quad (31)$$

### 2.3.2 Discussion

Comparing with section 2.1, we see that two Bayesian CMC estimators are indeed available: the SIR-based one  $\tilde{\Theta}_n^{\text{SIR}}$  given by (19), and the FA-based one  $\tilde{\Theta}_n^{\text{FA}}$  given by (31). The natural question which arises at this point is thus to wonder which one is best. Two arguments are available.

Let us first start from a common MC approximation  $\hat{p}_{0:n-1|n-1} = \sum_{i=1}^N w_{n-1}^i \delta_{\mathbf{x}_{0:n-1}^i}$  of  $p_{0:n-1|n-1}$ . Given  $\hat{p}_{0:n-1|n-1}$  and  $\mathbf{y}_n$ , trajectories  $\{\tilde{\mathbf{x}}_{0:n-1}^i\}_{i=1}^N$  produced by the FA algorithm are i.i.d. from  $\sum_{i=1}^N \tilde{w}_n^i \delta_{\mathbf{x}_{0:n-1}^i}$ . As is well known, resampling introduces variance, so given  $\hat{p}_{0:n-1|n-1}$   $\tilde{\Theta}_n^{\text{SIR}}$  is preferable to  $\tilde{\Theta}_n^{\text{FA}}$ , and  $\tilde{\Theta}_n^{\text{FA}}$  should not be used in practice.

On the other hand, the performances of  $\tilde{\Theta}_n^{\text{SIR}}$  also depend on the weighted trajectories  $\{(\mathbf{x}_{0:n-1}^i, w_{n-1}^i)\}_{i=1}^N$  which are available at time  $n-1$ ; so one can wonder whether one should propagate them via the SIR algorithm, or via the FA one.

This actually is a thorny issue, because in the SIR algorithm the resampling step is optional and is often performed according to a particular criterion, like an estimator of the so-called number of efficient particles [16] [17]. So comparing the set  $\{\tilde{\mathbf{x}}_n^i, \tilde{w}_n^i\}_{i=1}^N$  produced by the SIR algorithm *before* the resampling step to that  $\{\mathbf{x}_n^i, 1/N\}_{i=1}^N$  produced by the FA algorithm, is a challenging task, and indeed it has been proved in [22] (from an asymptotical point of view) that none algorithm always outperforms the other.

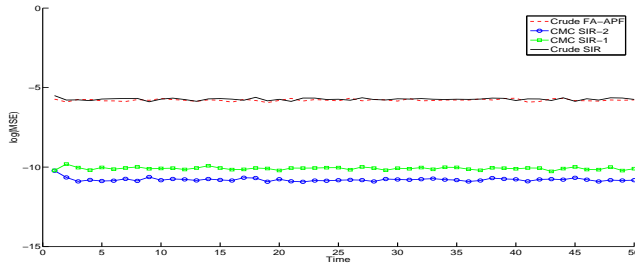
If however we assume that the resampling step is done at each time step, then it is well known [23, Ch. 9] that the set of samples produced by the FA algorithm is better (in an asymptotic normality sense) than that produced by the SIR algorithm *after* the resampling step. This can be easily understood empirically from a simple argument. Starting from a set of weighted samples  $\{\mathbf{x}_{0:n-1}^i, w_{n-1}^i\}_{i=1}^N$ , the number of different particles  $\{\mathbf{x}_n^i\}$  produced by the FA algorithm is equal to  $N$ , while that produced by the SIR one (after resampling) is lower than  $N$ , and can consequently lead to a poor approximation of  $p_{n|n}$ .

## 2.4 Simulations

In section 2.4.1 we compare via simulations two Bayesian CMC estimators  $\tilde{\Theta}_n^{\text{SIR}}$ , which differ only by the set of weighted points  $\hat{p}_{0:n-1|n-1}$  upon which they rely at each time instant  $n-1$ : this set will be either propagated by the SIR algorithm ( $\tilde{\Theta}_n^{\text{SIR},1}$ ), or by the FA one ( $\tilde{\Theta}_n^{\text{SIR},2}$ ). In section 2.4.2 we compare  $\tilde{\Theta}_n^{\text{FA}}$  and  $\tilde{\Theta}_n^{\text{SIR},2}$  in the ARCH model. In section 2.4.3 we compare the two approximations of  $\tilde{\Theta}_n$  described in section 2.2.2, and the weighted trajectories are propagated by the SIR algorithm. We compute the empirical mean square error (MSE) at each time step, averaged on  $P = 200$  simulations; the true mean is computed by the Kalman filter (KF) in the Gaussian case, or a bootstrap filter [24] with  $N = 10^5$  particles otherwise.

### 2.4.1 Gaussian Model

We first consider a linear and Gaussian model described by (20)-(21) where  $f_n(x_{n-1}) = 0.9x_{n-1}$ ,  $H_n = 1$ ,  $k_n(x_{n-1}) = \sqrt{10}$  and  $R_n^v = 1$ . We want to estimate the hidden state, so  $f(x_n) = x_n$ . We compute the SIR- and FA-based Bayesian crude and CMC estimators with  $N = 1000$  particles; of course KF, which computes  $E_{p_{n|n}}(x)$  exactly, is here the benchmark solution. MSEs of the four estimators are displayed in Fig. 1.  $\tilde{\Theta}_n^{\text{SIR},1}$  (resp.  $\tilde{\Theta}_n^{\text{SIR},2}$ ) always outperforms  $\hat{\Theta}_n^{\text{SIR}}$  (resp.  $\hat{\Theta}_n^{\text{FA}}$ ). Note also that  $\hat{\Theta}_n^{\text{FA}}$  does not always outperform  $\hat{\Theta}_n^{\text{SIR}}$ , which is in accordance with the asymptotical analysis [22]; while  $\tilde{\Theta}_n^{\text{SIR},2}$  always outperforms  $\tilde{\Theta}_n^{\text{SIR},1}$ .



**Fig. 1** MSE - Gaussian model,  $R = 1$ ,  $Q = 10$ ,  $N = 1000$  -  $f(x_n) = x_n$ . Estimator CMC-SIR-2 which propagates the samples with an FA algorithm outperforms CMC-SIR-1 which uses a SIR algorithm. Both CMC estimators outperform the crude ones.

### 2.4.2 ARCH Model

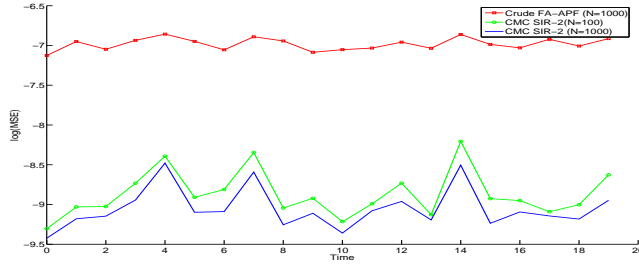
We next consider the ARCH model recalled in section 2.2.1 We set  $R_n^v = 3$ ,  $\beta_0 = 1$  and  $\beta_1 = 0.1$ . We want to estimate  $x_n$  (so  $f(x_n) = x_n$ ), and the variance of the process noise (so  $f(x_n) = \beta_0 + \beta_1 x_n^2$ ). Since  $p(x_n|x_{n-1}, y_n)$  is Gaussian (see (25)), it is possible to calculate both moments. We compare  $\hat{\Theta}_n^{\text{FA}}(1000)$  and  $\tilde{\Theta}_n^{\text{SIR},2}(1000)$ , both computed with  $N = 1000$  particles, and  $\tilde{\Theta}_n^{\text{SIR},2}(100)$ . MSEs are displayed on Fig. 2 for the estimate of  $x_n$  and Fig. 3 for the variance of the process noise. As we see  $\tilde{\Theta}_n^{\text{SIR},2}(1000)$ , and even  $\tilde{\Theta}_n^{\text{SIR},2}(100)$ , both outperform  $\hat{\Theta}_n^{\text{FA}}(1000)$ . However the gap between the three algorithms is function dependent and so the previous considerations are model and function dependent.

### 2.4.3 Stochastic Volatility Model

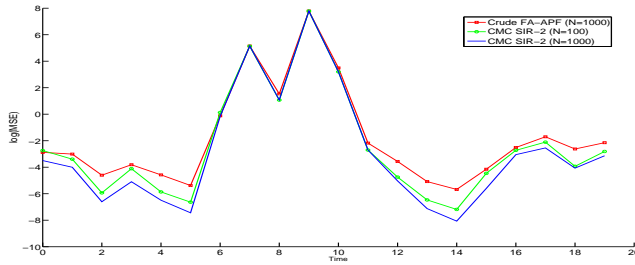
Let us consider the following model:

$$X_{n+1} = \Phi X_n + U_n \quad (32)$$

$$Y_n = \beta \exp(X_n/2) \times V_n \quad (33)$$



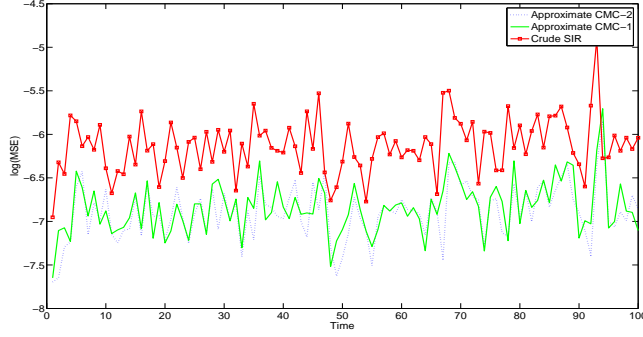
**Fig. 2** MSE - ARCH Model -  $\beta_0 = 1$ ,  $\beta_1 = 0.1$  and  $R_n^v = 3$  -  $f(x_n) = x_n$ . The CMC estimator with  $N = 100$  particles outperforms the crude one with  $N = 1000$  particles and is close to the CMC one with  $N = 1000$  particles.



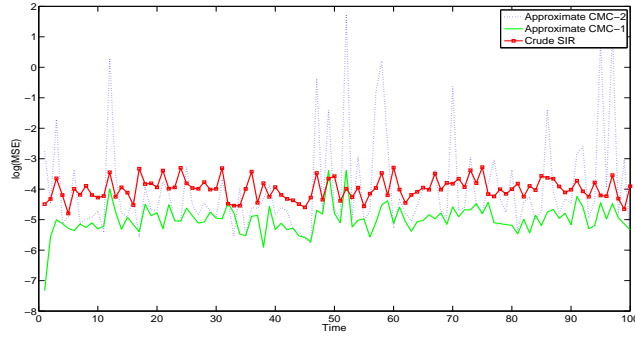
**Fig. 3** MSE - ARCH Model -  $\beta_0 = 1$ ,  $\beta_1 = 0.1$  and  $R_n^v = 3$  -  $f(x_n) = \beta_0 + \beta_1 x_{n-1}^2$ . The gap between the three estimators depends on function  $f(\cdot)$  and has decreased compared to the previous simulation.

in which  $U_n \sim \mathcal{N}(0, \sigma^2)$  and  $V_n \sim \mathcal{N}(0, 1)$ . In this model  $\tilde{\Theta}_n$  is not computable, whatever function  $f$ , because  $p(y_n|x_{n-1})$  is not computable. We propose to compare two approximations of the Bayesian CMC estimator with a SIR based crude estimator. Our first approximation  $\bar{\Theta}_n^{\text{SIR},1}$  only relies on the approximation of  $\int f(x_n)p(x_n|x_{n-1}, y_n)dx_n$  (second item in §2.2.2) while the second one  $\bar{\Theta}_n^{\text{SIR},2}$  relies in addition on that of  $p(y_n|x_{n-1})$  (first item in §2.2.2). In this model, an approximation of  $p(y_n|x_{n-1})$  is obtained by a first order Taylor series expansion of function  $\log(g_n(y_n|x_n))$  in  $\Phi x_{n-1}$ . If the deduced approximation of  $g_n(y_n|x_n)$  is noted  $\hat{g}_n(y_n|x_n)$  then  $\hat{p}(y_n|x_{n-1}) = \int f_{n|n-1}(x_n|x_{n-1})\hat{g}_n(y_n|x_n)dx_n$  where  $f_{n|n-1}(x_n|x_{n-1}) = \mathcal{N}(x_n; \Phi x_{n-1}; \sigma^2)$ , is now computable. If  $\sigma$  is small,  $f_{n|n-1}(x_n|x_{n-1})$  is approximately non-null for values close to  $\Phi x_{n-1}$ , and for such values  $\hat{g}_n(y_n|x_n)$  is a good approximation of  $g_n(y_n|x_n)$ . So one should get a good approximation  $\hat{p}(y_n|x_{n-1})$  when  $\sigma$  is small. Finally, a deduced approximation of  $p(x_n|x_{n-1}, y_n)$  is given by a Gaussian pdf, see [20].

We estimate the standard deviation of the observation noise at time  $n$  so  $f(x_n) = \beta \exp(x_n/2)$ . We first take  $\Phi = 0.8$ ,  $\beta = 0.6$ ,  $\sigma = 0.18$ . Results are displayed in Fig. 4. We observe that both approximations of the Bayesian CMC



**Fig. 4** MSE - Stochastic Volatility Model -  $\Phi = 0.8$ ,  $\beta = 0.6$ ,  $\sigma = 0.18$ ,  $N = 1000$ . Both approximate CMC estimators outperform the crude one.



**Fig. 5** MSE - Stochastic Volatility Model -  $\Phi = 0.8$ ,  $\beta = 0.6$ ,  $\sigma = 0.4$ ,  $N = 1000$ . Only the approximate estimator CMC-1 outperforms the crude estimator. This is because the approximation of  $p(\mathbf{y}_n|\mathbf{x}_{n-1})$  used by the approximate estimator CMC-2 is not reliable with these parameters.

estimator outperform the crude SIR-based one, and that the second approximation, which does not use  $\mathbf{x}_n^{1:N}$ , is preferable. However, in Fig. 5 we take  $\sigma = 0.40$ . Remember that increasing  $\sigma$  has consequences on the approximation of  $p(y_n|x_{n-1})$ ; as expected,  $\bar{\Theta}_n^{\text{SIR},2}$ , which relies on this approximation, is outperformed by the two other estimators. It is particularly interesting to notice that the first approximation  $\bar{\Theta}_n^{\text{SIR},1}$  is not affected and still outperforms the SIR based estimator. This confirms that Bayesian CMC estimators can still be of practical interest in models which are not semi-linear.

### 3 Bayesian CMC algorithms for JMSS models

As in §2 we still consider the estimation of  $\Theta_n = \int \phi(\mathbf{x}_n)p(\mathbf{x}_n|\mathbf{y}_{0:n})d\mathbf{x}_n$  (the reason why we replaced  $f$  by  $\phi$  will become clear a few lines below), but now

in a so-called JMSS:

$$p(\mathbf{r}_{0:n}, \mathbf{x}_{0:n}, \mathbf{y}_{0:n}) = p(r_0) \prod_{i=1}^n p(r_i | r_{i-1}) p(\mathbf{x}_0) \prod_{i=1}^n f_{i|i-1}(\mathbf{x}_i | \mathbf{x}_{i-1}, r_i) \prod_{i=0}^n g_i(\mathbf{y}_i | \mathbf{x}_i, r_i). \quad (34)$$

Model (34) can be thought of as an HMC model, in which  $f_{i|i-1}$  and  $g_i$  depend on the realization of a discrete Markov Chain  $\{R_n\}_{n \geq 0}$  where each  $R_n$  takes its values in  $\{1, \dots, K\}$ . So now both  $X_n$  and  $R_n$  are hidden, and  $\Theta_n$  can be rewritten as

$$\Theta_n = \sum_{\mathbf{r}_{0:n}} \int \phi(\mathbf{x}_n) p(\mathbf{x}_n, \mathbf{r}_{0:n} | \mathbf{y}_{0:n}) d\mathbf{x}_n. \quad (35)$$

Note that  $\phi(\cdot)$  can also depend on  $r_n$ . As is well known [25] [26] [7], in a JMSS exact Bayesian filtering is either impossible (in the general case) or an NP-hard problem (in the linear and Gaussian case), so one has to use suboptimal techniques. Among them, SMC methods can be divided into two classes:

- In the first class [27] [28] [23]  $\Theta_n$  is computed by injecting an SMC approximation of  $p(\mathbf{x}_{0:n}, \mathbf{r}_{0:n} | \mathbf{y}_{0:n})$  into (35);
- In the second class of SMC methods we start from

$$\Theta_n = \sum_{\mathbf{r}_{0:n}} \underbrace{p(\mathbf{r}_{0:n} | \mathbf{y}_{0:n})}_{\text{PF}} \int \phi(\mathbf{x}_n) \underbrace{p(\mathbf{x}_n | \mathbf{r}_{0:n}, \mathbf{y}_{0:n})}_{\text{KF}} d\mathbf{x}_n, \quad (36)$$

and propagate an SMC approximation  $\sum_{i=1}^N w_n^i \delta_{\mathbf{r}_{0:n}^i}$  of  $p(\mathbf{r}_{0:n} | \mathbf{y}_{0:n})$  only; then  $\hat{\Theta}_n$  is computed as

$$\hat{\Theta}_n = \sum_{i=1}^N w_n^i \int \phi(\mathbf{x}_n) \underbrace{p(\mathbf{x}_n | \mathbf{r}_{0:n}^i, \mathbf{y}_{0:n})}_{\text{KF}} d\mathbf{x}_n, \quad (37)$$

in which  $p(\mathbf{x}_n | \mathbf{r}_{0:n}^i, \mathbf{y}_{0:n})$  is computed exactly via KF if model (34), conditionally on  $\mathbf{r}_{0:n}$ , is linear and Gaussian, i.e. if  $f_{i+1|i}(\mathbf{x}_{i+1} | \mathbf{x}_i, r_{i+1})$  and  $g_i(\mathbf{y}_i | \mathbf{x}_i, r_i)$  are Gaussian with means linear in  $\mathbf{x}_i$  [7].

### 3.1 Bayesian CMC algorithms for linear and Gaussian JMSS models

#### 3.1.1 Deriving the Bayesian CMC algorithm

In this section we begin with the second class of algorithms. Let us first see that (36) coincides with (6) (in which the integral is replaced by a sum, since  $R_n$  is discrete), up to the identification:  $X_1 = \mathbf{R}_{0:n-1}$ ,  $X_2 = R_n$ ,  $f(x_1, x_2) = \int \phi(\mathbf{x}_n) p(\mathbf{x}_n | \mathbf{r}_{0:n-1}, r_n, \mathbf{y}_{0:n}) d\mathbf{x}_n$ , and  $p(x_1, x_2)$  is the joint pdf

$$p(\mathbf{r}_{0:n} | \mathbf{y}_{0:n}) = \underbrace{p(\mathbf{r}_{0:n-1} | \mathbf{y}_{0:n})}_{p(x_1)} \underbrace{p(r_n | \mathbf{r}_{0:n-1}, \mathbf{y}_{0:n})}_{p(x_2 | x_1)}. \quad (38)$$

We need to compute both factors (we cannot simply apply the results of §2.1, because in (34) the marginal chain  $(R_n, \mathbf{Y}_n)$  is not an HMC, as was  $(\mathbf{X}_n, \mathbf{Y}_n)$  in (13)). We first need an approximation of  $p(x_1)$ , i.e. of

$$p(\mathbf{r}_{0:n-1}|\mathbf{y}_{0:n}) = \frac{p(\mathbf{y}_n|\mathbf{r}_{0:n-1}, \mathbf{y}_{0:n-1})p(\mathbf{r}_{0:n-1}|\mathbf{y}_{0:n-1})}{\sum_{\mathbf{r}_{0:n-1}} p(\mathbf{y}_n|\mathbf{r}_{0:n-1}, \mathbf{y}_{0:n-1})p(\mathbf{r}_{0:n-1}|\mathbf{y}_{0:n-1})}. \quad (39)$$

However the SMC algorithm propagates approximations of  $p(\mathbf{r}_{0:n}|\mathbf{y}_{0:n})$ . So let  $\hat{p}(\mathbf{r}_{0:n-1}|\mathbf{y}_{0:n-1}) = \sum_{i=1}^N w_{n-1}^i \delta_{\mathbf{r}_{0:n-1}^i}$ ; applying again Rubin's SIR mechanism,  $\hat{p}(\mathbf{r}_{0:n-1}|\mathbf{y}_{0:n}) = \sum_{i=1}^N \tilde{w}_{n-1}^i \delta_{\mathbf{r}_{0:n-1}^i}$ , where

$$\tilde{w}_{n-1}^i \propto w_{n-1}^i p(\mathbf{y}_n|\mathbf{y}_{0:n-1}, \mathbf{r}_{0:n-1}^i), \quad \sum_{i=1}^N \tilde{w}_{n-1}^i = 1, \quad (40)$$

is an MC approximation of  $p(\mathbf{r}_{0:n-1}|\mathbf{y}_{0:n})$ . Next from (34), the second factor  $p(x_2|x_1)$  of (38) can be rewritten as (here  $\mathcal{N}$  stands for numerator):

$$p(r_n|\mathbf{r}_{0:n-1}^i, \mathbf{y}_{0:n}) = \frac{p(\mathbf{y}_n|\mathbf{y}_{0:n-1}, \mathbf{r}_{0:n-1}^i, r_n)p(r_n|r_{n-1}^i)}{p(\mathbf{y}_n|\mathbf{y}_{0:n-1}, \mathbf{r}_{0:n-1}^i) = \sum_{r_n} \mathcal{N}}. \quad (41)$$

Note that as in section 2,  $p(r_n|\mathbf{r}_{0:n-1}^i, \mathbf{y}_{0:n})$  is the optimal conditional IS distribution, i.e. that which minimizes the conditional variance of the weights, given  $\mathbf{r}_{0:n-1}^i$  and  $\mathbf{y}_{0:n}$ . Finally, setting  $f(\mathbf{r}_{0:n-1}^i, r_n) = E(\phi(\mathbf{x}_n)|\mathbf{y}_{0:n}, \mathbf{r}_{0:n-1}^i, r_n)$ , the Bayesian CMC and crude estimators respectively read

$$\tilde{\Theta}_n(\mathbf{r}_{0:n-1}^{1:N}) = \sum_{i=1}^N \tilde{w}_{n-1}^i(\mathbf{r}_{0:n-1}^{1:N}) \sum_{r_n} f(\mathbf{r}_{0:n-1}^i, r_n) p(r_n|\mathbf{r}_{0:n-1}^i, \mathbf{y}_{0:n}), \quad (42)$$

$$\hat{\Theta}_n(\mathbf{r}_{0:n-1}^{1:N}, \mathbf{r}_n^{1:N}) = \sum_{i=1}^N \tilde{w}_{n-1}^i(\mathbf{r}_{0:n-1}^{1:N}) f(\mathbf{r}_{0:n-1}^i, r_n^i), \quad (43)$$

in which  $r_n^i \sim p(r_n|\mathbf{r}_{0:n-1}^i, \mathbf{y}_{0:n})$ .

### 3.1.2 Computing $\tilde{\Theta}_n$ in practice: linear and Gaussian JMSS

Implementing  $\tilde{\Theta}_n$  requires that (40) and (41) are computable, and that in (42) the conditional expectation  $f(\mathbf{r}_{0:n-1}^i, r_n)$  is computable too. We thus need to compute  $p(\mathbf{y}_n|\mathbf{y}_{0:n-1}, \mathbf{r}_{0:n-1}^i, r_n)$ , which is not possible in general JMSS models. So let us now assume that the JMSS (34) is moreover linear and (conditionally) Gaussian:

$$R_n \text{ is a discrete Markov Chain,} \quad (44)$$

$$\mathbf{X}_n = \mathbf{F}_n(R_n)\mathbf{X}_{n-1} + \mathbf{G}_n(R_n)\mathbf{U}_n, \quad (45)$$

$$\mathbf{Y}_n = \mathbf{H}_n(R_n)\mathbf{X}_n + \mathbf{L}_n(R_n)\mathbf{V}_n, \quad (46)$$

where  $\mathbf{X}_0, \mathbf{U}_1, \dots, \mathbf{U}_n$  and  $\mathbf{V}_0, \dots, \mathbf{V}_n$  are independent and independent of  $R_0, \dots, R_n$ . We set  $\mathbf{X}_0 \sim \mathcal{N}(\mathbf{m}_0, \mathbf{P}_0)$ ,  $\mathbf{U}_n \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_n)$  and  $\mathbf{V}_n \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_n^v)$ .

Then let  $p(\mathbf{x}_{n-1}|\mathbf{r}_{0:n-1}^i, \mathbf{y}_{0:n-1}) = \mathcal{N}(\mathbf{x}_{n-1}; \mathbf{m}_{n-1|n-1}^i; \mathbf{P}_{n-1|n-1}^i)$ . Then  $p(\mathbf{y}_n|\mathbf{y}_{0:n-1}, \mathbf{r}_{0:n-1}^i, r_n)$  is given by the predicted observation mean and covariance of the KF, i.e.

$$p(\mathbf{y}_n|\mathbf{y}_{0:n-1}, \mathbf{r}_{0:n-1}^i, r_n) = \mathcal{N}(\mathbf{y}_n; \tilde{\mathbf{y}}_n^i(r_n); \mathbf{S}_n^i(r_n)), \quad (47)$$

where

$$\tilde{\mathbf{y}}_n^i(r_n) = \mathbf{y}_n - \mathbf{H}_n(r_n)\mathbf{F}_n(r_n)\mathbf{m}_{n-1|n-1}^i, \quad (48)$$

$$\mathbf{S}_n^i(r_n) = \mathbf{H}_n(r_n)\mathbf{P}_{n|n-1}^i(r_n)\mathbf{H}_n(r_n)^T + \mathbf{L}_n(r_n)\mathbf{R}_n^v\mathbf{L}_n(r_n)^T \quad (49)$$

$$\mathbf{P}_{n|n-1}^i(r_n) = \mathbf{F}_n(r_n)\mathbf{P}_{n-1|n-1}^i\mathbf{F}_n^T(r_n) + \mathbf{G}_n(r_n)\mathbf{Q}_n\mathbf{G}_n(r_n)^T. \quad (50)$$

In summary, (47)-(50) enable to compute (40) and (41), and finally (42).

*Remark 2* Estimator  $\tilde{\Theta}_n$  in (42) is the Bayesian CMC counterpart of  $\hat{\Theta}_n$  in (43), which itself coincides with the so-called RB SMC estimator (37) for JMSS [7]. Indeed,  $\hat{\Theta}_n$  corresponds to  $\hat{\Theta}^{\text{RB}}$  in (5) where  $\mathbf{x}_1 = \mathbf{r}_{0:n}$ ,  $\mathbf{x}_2 = \mathbf{x}_{0:n}$ ,  $f(\mathbf{x}_1, \mathbf{x}_2) = \phi(\mathbf{x}_n)$  and  $q_1(r_n|\mathbf{r}_{0:n-1}) = p(r_n|\mathbf{r}_{0:n-1}, \mathbf{y}_{0:n})$ . So (42) can be seen as a further RB step of an already RB SMC estimator; the RB step leading to (37) was a spatial one, since PF was performed on variables  $\mathbf{r}_{0:n}$ , rather than on the extended state  $(\mathbf{x}_{0:n}, \mathbf{r}_{0:n})$ ; here this second RB step is temporal, since in (42) PF acts on  $\mathbf{r}_{0:n-1}$ , rather than on  $\mathbf{r}_{0:n}$ . So here is an example where we can jointly use the classical RB-PF and our CMC Bayesian technique; but we will see in the next section that a CMC Bayesian estimator can also be derived in JMSS models in which classical RB-PF is not available.

*Remark 3* One should observe that if  $\hat{\Theta}_n$  can be computed,  $\tilde{\Theta}_n$  can be computed as well; so the variance reduction can be achieved under the same assumptions (linear and Gaussian JMSS) as those needed for the RB SMC estimator [7]. On the other hand this new variance reduction involves an extra computational effort, which however is not prohibitive (at least if  $K$  is small), as we see from (42) and (43). First, weights  $\tilde{w}_{n-1}^i$  in (40) have to be computed by both algorithms. Next for each  $i$ ,  $1 \leq i \leq N$ , both algorithms compute  $\{p(r_n|\mathbf{r}_{0:n-1}^i, \mathbf{y}_{0:n})\}_{r_n=1}^K$ . The difference is that in the CMC algorithm we compute directly means  $\sum_{r_n} f(\mathbf{r}_{0:n-1}^i, r_n)p(r_n|\mathbf{r}_{0:n-1}^i, \mathbf{y}_{0:n})$ , which requires running  $K$  KF updating steps per trajectory  $\mathbf{r}_{0:n-1}^i$  while the crude estimator first extends randomly each trajectory before computing conditional expectations.

*Remark 4* Finally our Bayesian CMC estimator  $\tilde{\Theta}_n$  stems from the RB PF which, itself, assumes that the JMSS model is conditionally linear with additive Gaussian noise. If this is not the case, but the non-linearities are not too severe, one can approximate  $E(\phi(\mathbf{x}_n)|\mathbf{y}_{0:n}, \mathbf{r}_{0:n})$  by EKF or UKF, and next compute  $\tilde{\Theta}_n$  from such an approximation, by using an approximation of  $p(\mathbf{y}_n|\mathbf{y}_{0:n-1}, \mathbf{r}_{0:n})$ , also given by EKF or UKF.



### 3.2 Bayesian CMC algorithms for non linear JMSS models

In this section we derive Bayesian CMC estimators in non linear JMSS models, in the case where, by contrast with Remark 4 above, it is not possible to approximate  $E(\phi(\mathbf{x}_n)|\mathbf{y}_{0:n}, \mathbf{r}_{0:n})$ . In that case we need to turn back to the first class of SMC methods for JMSS (see the beginning of section 3), which consists in propagating an SMC approximation of  $p(\mathbf{x}_{0:n}, \mathbf{r}_{0:n}|\mathbf{y}_{0:n})$  [7] [29].

#### 3.2.1 Deriving Bayesian CMC estimators

Let us first rewrite  $\Theta_n$  as

$$\Theta_n = \sum_{\mathbf{r}_{0:n-1}, r_n} \int \phi(\mathbf{x}_n) p(\mathbf{x}_{0:n-1}, \mathbf{r}_{0:n-1}, \mathbf{x}_n, r_n | \mathbf{y}_{0:n}) d\mathbf{x}_{0:n-1} d\mathbf{x}_n. \quad (51)$$

Let now  $\sum_{i=1}^N w_{n-1}^i(\mathbf{x}_{0:n-1}^{1:N}, \mathbf{r}_{0:n-1}^{1:N}) \delta_{\mathbf{x}_{0:n-1}^i, \mathbf{r}_{0:n-1}^i}$  be an MC approximation of  $p(\mathbf{x}_{0:n-1}, \mathbf{r}_{0:n-1} | \mathbf{y}_{0:n-1})$ . Then  $\hat{p}(\mathbf{x}_{0:n-1}, \mathbf{r}_{0:n-1} | \mathbf{y}_{0:n}) = \sum_{i=1}^N \tilde{w}_{n-1}^i \delta_{\mathbf{x}_{0:n-1}^i, \mathbf{r}_{0:n-1}^i}$ , in which

$$\tilde{w}_{n-1}^i \propto w_{n-1}^i p(\mathbf{y}_n | \mathbf{x}_{n-1}^i, r_{n-1}^i), \sum_{i=1}^N \tilde{w}_{n-1}^i = 1 \quad (52)$$

is an MC approximation of  $p(\mathbf{x}_{0:n-1}, \mathbf{r}_{0:n-1} | \mathbf{y}_{0:n})$ . Let also

$$(\mathbf{x}_n^i, r_n^i) \sim p(\mathbf{x}_n, r_n | \mathbf{x}_{n-1}^i, r_{n-1}^i, \mathbf{y}_n). \quad (53)$$

Then the associated crude MC estimator is given by [7] [29]:

$$\hat{\Theta}_n(\mathbf{x}_{0:n}^{1:N}, \mathbf{r}_{0:n}^{1:N}) = \sum_{i=1}^N \tilde{w}_{n-1}^i(\mathbf{x}_{0:n-1}^{1:N}, \mathbf{r}_{0:n-1}^{1:N}) \phi(\mathbf{x}_n^i), \quad (54)$$

We now propose two Bayesian CMC estimators of  $\Theta_n$ , associated to two different partitions of  $(\mathbf{X}_{0:n}, \mathbf{R}_{0:n})$ . Setting  $X_1 = (\mathbf{X}_{0:n-1}, \mathbf{R}_{0:n})$  and  $X_2 = \mathbf{X}_n$  leads to  $\tilde{\Theta}_n^{\mathbf{X}_n}$ ; setting  $X_1 = (\mathbf{X}_{0:n-1}, \mathbf{R}_{0:n-1})$ ,  $X_2 = (\mathbf{X}_n, R_n)$  and  $f(\mathbf{x}_{n-1}^i, \mathbf{y}_n, \mathbf{r}_n) = \int \phi(\mathbf{x}_n) p(\mathbf{x}_n | \mathbf{x}_{n-1}^i, r_n, \mathbf{y}_n) d\mathbf{x}_n$  leads to  $\tilde{\Theta}_n^{(\mathbf{X}_n, R_n)}$ , with

$$\tilde{\Theta}_n^{\mathbf{X}_n}(\mathbf{x}_{0:n-1}^{1:N}, \mathbf{r}_{0:n}^{1:N}) = \sum_{i=1}^N \tilde{w}_{n-1}^i f(\mathbf{x}_{n-1}^i, \mathbf{y}_n, \mathbf{r}_n^i), \quad (55)$$

$$\begin{aligned} \tilde{\Theta}_n^{(\mathbf{X}_n, R_n)}(\mathbf{x}_{0:n-1}^{1:N}, \mathbf{r}_{0:n-1}^{1:N}) &= \sum_{i=1}^N \tilde{w}_{n-1}^i \sum_{r_n} \int \phi(\mathbf{x}_n) p(\mathbf{x}_n, r_n | \mathbf{x}_{n-1}^i, r_{n-1}^i, \mathbf{y}_n) d\mathbf{x}_n \\ &= \sum_{i=1}^N \tilde{w}_{n-1}^i \sum_{r_n} p(r_n | \mathbf{x}_{n-1}^i, r_{n-1}^i, \mathbf{y}_n) f(\mathbf{x}_{n-1}^i, \mathbf{y}_n, r_n), \end{aligned} \quad (56)$$

in which  $\tilde{w}_{n-1}^i(\mathbf{x}_{0:n-1}^{1:N}, \mathbf{r}_{0:n-1}^{1:N})$  is given by (52), and in (55)  $r_n^i \sim p(r_n | \mathbf{x}_{n-1}^i, r_{n-1}^i, \mathbf{y}_n)$  (a marginal of (53)). Of course,  $\tilde{\Theta}_n^{\mathbf{X}_n} = E(\hat{\Theta}_n | \mathbf{x}_{0:n-1}^{1:N}, \mathbf{r}_{0:n-1}^{1:N}, \mathbf{y}_{0:n})$  and  $\tilde{\Theta}_n^{(\mathbf{X}_n, R_n)} = E(\tilde{\Theta}_n^{\mathbf{X}_n} | \mathbf{x}_{0:n-1}^{1:N}, \mathbf{r}_{0:n-1}^{1:N}, \mathbf{y}_{0:n})$ , so  $\text{var}(\tilde{\Theta}_n^{(\mathbf{X}_n, R_n)}) \leq \text{var}(\tilde{\Theta}_n^{\mathbf{X}_n}) \leq \text{var}(\hat{\Theta}_n)$ .

### 3.2.2 Computing $\tilde{\Theta}_n^{\mathbf{X}_n}$ and $\tilde{\Theta}_n^{(\mathbf{X}_n, R_n)}$ in practice

Let us now discuss when (55) and (56) can be computed. In model (34),  $p(\mathbf{y}_n | \mathbf{x}_{n-1}^i, r_{n-1}^i) = \sum_{r_n} p(r_n | r_{n-1}^i) p(\mathbf{y}_n | \mathbf{x}_{n-1}^i, r_n)$  and  $p(\mathbf{y}_n | \mathbf{x}_{n-1}^i, r_{n-1}^i) p(\mathbf{x}_n, r_n | \mathbf{x}_{n-1}^i, r_{n-1}^i, \mathbf{y}_n) = p(r_n | r_{n-1}^i) p(\mathbf{y}_n | \mathbf{x}_{n-1}^i, r_n) p(\mathbf{x}_n | \mathbf{x}_{n-1}^i, r_n, \mathbf{y}_n)$ . So let

$$\bar{w}_{n-1}^i(r_n) = \frac{w_{n-1}^i p(r_n | r_{n-1}^i) p(\mathbf{y}_n | \mathbf{x}_{n-1}^i, r_n)}{\sum_{r_n} \sum_{i=1}^N w_{n-1}^i p(r_n | r_{n-1}^i) p(\mathbf{y}_n | \mathbf{x}_{n-1}^i, r_n)} \quad (57)$$

(note that  $\sum_{r_n} \bar{w}_{n-1}^i(r_n) = \tilde{w}_{n-1}^i$ ). Then  $\tilde{\Theta}_n^{\mathbf{X}_n}$  and  $\tilde{\Theta}_n^{(\mathbf{X}_n, R_n)}$  can be rewritten as

$$\tilde{\Theta}_n^{\mathbf{X}_n} = \sum_{i=1}^N \left[ \sum_{r_n} \bar{w}_{n-1}^i(r_n) \right] \int \phi(\mathbf{x}_n) p(\mathbf{x}_n | \mathbf{x}_{n-1}^i, \mathbf{y}_n, r_n^i) d\mathbf{x}_n, \quad (58)$$

$$\tilde{\Theta}_n^{(\mathbf{X}_n, R_n)} = \sum_{i=1}^N \left[ \sum_{r_n} \bar{w}_{n-1}^i(r_n) \int \phi(\mathbf{x}_n) p(\mathbf{x}_n | \mathbf{x}_{n-1}^i, \mathbf{y}_n, r_n) d\mathbf{x}_n \right]. \quad (59)$$

So (59) is computable as soon as (58) is computable. On the other hand,  $\tilde{\Theta}_n^{\mathbf{X}_n}$  is a generalization of (19):  $p(\mathbf{x}_n | \mathbf{x}_{n-1}, \mathbf{y}_n, r_n)$  and  $p(\mathbf{y}_n | \mathbf{x}_{n-1}, r_n)$  play the same role as  $p(\mathbf{x}_n | \mathbf{x}_{n-1}, \mathbf{y}_n)$  and  $p(\mathbf{y}_n | \mathbf{x}_{n-1})$  in §2.1 except that we have now introduced a dependency in  $r_n$ . This means that  $\tilde{\Theta}_n^{\mathbf{X}_n}$  and  $\tilde{\Theta}_n^{(\mathbf{X}_n, R_n)}$  are computable as soon as the Bayesian CMC estimator (19) of §2.1 is computable in the underlying HMC model (i.e., the HMC model to which the JMSS reduces when the jumps are known), see section 2.2.1. For example, semi-linear stochastic models (including the ARCH ones) with Markov jumps are a class of models in which (58) and (59) are computable.

Finally the only difference between (58) and (59) comes from the computational cost that we discuss now. For a given  $i$ , in (59) the computation of  $\int \phi(\mathbf{x}_n) p(\mathbf{x}_n | \mathbf{x}_{n-1}^i, r_n, \mathbf{y}_n) d\mathbf{x}_n$  has to be done for all  $r_n$ , while in (58) it has only to be done for the  $r_n^i$  which has been sampled. So as expected,  $\tilde{\Theta}_n^{(\mathbf{X}_n, R_n)}$  is preferable to  $\tilde{\Theta}_n^{\mathbf{X}_n}$  but requires an extra computational cost. On the other hand, comparing the computational cost of  $\tilde{\Theta}_n^{\mathbf{X}_n}$  and  $\hat{\Theta}_n$  is the same issue as comparing that of the Bayesian CMC estimator (19) to that of the crude MC one (18) in Section 2.2, and is thus problem dependent. However, one should observe that in the particular case described at the end of section 2.2.1, i.e. when sampling according to  $p(\mathbf{x}_n | \mathbf{x}_{n-1}, r_n, \mathbf{y}_n)$  requires the computation of  $\int \phi(\mathbf{x}_n) p(\mathbf{x}_n | \mathbf{x}_{n-1}, r_n, \mathbf{y}_n) d\mathbf{x}_n$ , then the computation of  $\tilde{\Theta}_n^{\mathbf{X}_n}$  does not involve an extra computational cost as compared to that of  $\hat{\Theta}_n$ .

### 3.2.3 Approximate computation

Let us finally discuss on approximate computation of  $\tilde{\Theta}_n$  when  $p(\mathbf{y}_n | \mathbf{x}_{n-1}, r_n)$  and  $p(\mathbf{x}_n | \mathbf{x}_{n-1}, \mathbf{y}_n, r_n)$  in (59) are not available. First notice that (59) can be computed with the same numerical approximations as those which were used

in the computation of (19) (see section 2.2.2 above), except that they have to be done for all possible values of  $r_n$ . However,  $r_n$  is discrete and as we now see, one can derive other approximation techniques:

- In (34) we have  $p(r_n, \mathbf{x}_n, \mathbf{y}_n | \mathbf{x}_{n-1}, r_{n-1}) = p(r_n | r_{n-1}) f_{n|n-1}(\mathbf{x}_n | \mathbf{x}_{n-1}, r_n) g_n(\mathbf{y}_n | \mathbf{x}_n, r_n)$  so the numerator of (59) can be rewritten as

$$\sum_{r_n} \int \phi(\mathbf{x}_n) \left[ \sum_{i=1}^N p(r_n | r_{n-1}^i) w_{n-1}^i f_{n|n-1}(\mathbf{x}_n | \mathbf{x}_{n-1}^i, r_n) g_n(\mathbf{y}_n | \mathbf{x}_n, r_n) \right] d\mathbf{x}_n.$$

If for a given  $r_n$  the integral is not computable, one can approximate it with IS by sampling  $\mathbf{x}_n^{r_n, i} \sim q(\mathbf{x}_n | \mathbf{x}_{n-1}^i, r_n)$  for all  $i$ ,  $1 \leq i \leq N$  and for all  $r_n$ . An approximation of the numerator is then given by  $\sum_{r_n} \sum_{i=1}^N \phi(\mathbf{x}_n^{r_n, i}) p(r_n | r_{n-1}^i) w_{n-1}^i f_{n|n-1}(\mathbf{x}_n^{r_n, i} | \mathbf{x}_{n-1}^i, r_n) g_n(\mathbf{y}_n | \mathbf{x}_n^{r_n, i}, r_n) / q(\mathbf{x}_n^{r_n, i} | \mathbf{x}_{n-1}^i, r_n)$ . So we do not use samples for the discrete part  $r_n$ . We apply the same approximation for the denominator which can be rewritten as  $\sum_{r_n} \sum_{i=1}^N w_{n-1}^i p(r_n | r_{n-1}^i) \times \int f_{n|n-1}(\mathbf{x}_n | r_n, \mathbf{x}_{n-1}^i) g_n(\mathbf{y}_n | \mathbf{x}_n, r_n) d\mathbf{x}_n$ ;

- When the optimal distribution  $p(\mathbf{x}_n, r_n | \mathbf{x}_{n-1}, r_{n-1}, \mathbf{y}_n)$  is not available, it has been proposed [7] to sample independently  $(\mathbf{x}_n^i, r_n^i)$  according to an importance distribution  $q(\mathbf{x}_n, r_n | \mathbf{x}_{n-1}, r_{n-1}) = q(r_n | \mathbf{x}_{n-1}, r_{n-1}) q(\mathbf{x}_n | \mathbf{x}_{n-1}, r_n, r_{n-1})$ , for all  $i$ ,  $1 \leq i \leq N$ , then to compute the estimator  $\hat{\Theta}_n^{\text{SIR}} = \sum_{i=1}^N \frac{w_n^i}{\sum_{i=1}^N w_n^i} \phi(\mathbf{x}_n^i)$ ,  $w_n^i = w_{n-1}^i \frac{p(r_n^i | r_{n-1}^i) f_{n|n-1}(\mathbf{x}_n^i | r_n^i, \mathbf{x}_{n-1}^i) g_n(\mathbf{y}_n | \mathbf{x}_n^i, r_n^i)}{q(\mathbf{x}_n^i, r_n^i | \mathbf{x}_{n-1}^i, r_{n-1}^i)}$ . Remember that the Bayesian CMC estimator  $\hat{\Theta}_n$  is actually the expectation of the crude MC estimator  $\hat{\Theta}_n$  given some variables. One can wonder if it is not possible to compute  $E(\hat{\Theta}_n^{\text{SIR}} | \{r_{n-1}^i, \mathbf{x}_{n-1}^i, \mathbf{x}_n^i\}_{i=1}^N, \mathbf{y}_n)$ , i.e. to compute the expectation of  $\hat{\Theta}_n^{\text{SIR}}$  as a function of  $r_n$ . Since  $\phi(\cdot)$  does not depend on  $r_n$ , it is equivalent to compute the conditional expectation of the unnormalized weights  $w_n^i$  and so to reduce their variance. Unfortunately this is not possible because of the normalization factor. However one can compute separately the conditional expectation of the numerator and that of the denominator. This is an easy task since  $r_n$  takes its values in a discrete set, and  $q(r_n | \mathbf{x}_n, \mathbf{x}_{n-1}, r_{n-1}) = q(r_n | \mathbf{x}_{n-1}, r_{n-1}) q(\mathbf{x}_n | \mathbf{x}_{n-1}, r_n, r_{n-1}) / \sum_{r_n} q(r_n | \mathbf{x}_{n-1}, r_{n-1}) q(\mathbf{x}_n | \mathbf{x}_{n-1}, r_n, r_{n-1})$ . This variance reduction of the unnormalized weights comes from a normalized IS implementation of (59) which is rewritten as

$$\tilde{\Theta}_n = \frac{\int \phi(\mathbf{x}_n) \sum_{i=1}^N \left[ \sum_{r_n} p(r_n | r_{n-1}^i) w_{n-1}^i f_{n|n-1}(\mathbf{x}_n | \mathbf{x}_{n-1}^i, r_n) g_n(\mathbf{y}_n | \mathbf{x}_n, r_n) \right] d\mathbf{x}_n}{\int \sum_{i=1}^N \left[ \sum_{r_n} p(r_n | r_{n-1}^i) w_{n-1}^i f_{n|n-1}(\mathbf{x}_n | \mathbf{x}_{n-1}^i, r_n) g_n(\mathbf{y}_n | \mathbf{x}_n, r_n) \right] d\mathbf{x}_n} \quad (60)$$

with the importance distribution

$$q(\mathbf{x}_n | \mathbf{x}_{n-1}, r_{n-1}) = \sum_{r_n} q(r_n | \mathbf{x}_{n-1}, r_{n-1}) q(\mathbf{x}_n | \mathbf{x}_{n-1}, r_n, r_{n-1}).$$

Note that the computation of the new weights is not prohibitive as long as  $K \ll N$ .

### 3.3 Simulations

We now test our approach in a linear and Gaussian JMSS model, described by equations (44)-(46) in which

$$\mathbf{F}_n(r) = \begin{bmatrix} 1 & \frac{\sin(\omega_r T)}{\omega_r} & 0 & -\frac{1-\cos(\omega_r T)}{\omega_r} \\ 0 & \cos(\omega_r T) & 0 & -\sin(\omega_r T) \\ 0 & \frac{1-\cos(\omega_r T)}{\omega_r} & 1 & \frac{\sin(\omega_r T)}{\omega_r} \\ 0 & \sin(\omega_r T) & 0 & \cos(\omega_r T) \end{bmatrix}, \quad \mathbf{Q}_n = \sigma_v^2 \begin{bmatrix} \frac{T^3}{3} & \frac{T^2}{2} & 0 & 0 \\ \frac{T^2}{2} & T & 0 & 0 \\ 0 & 0 & \frac{T^3}{3} & \frac{T^2}{2} \\ 0 & 0 & \frac{T^2}{2} & T \end{bmatrix},$$

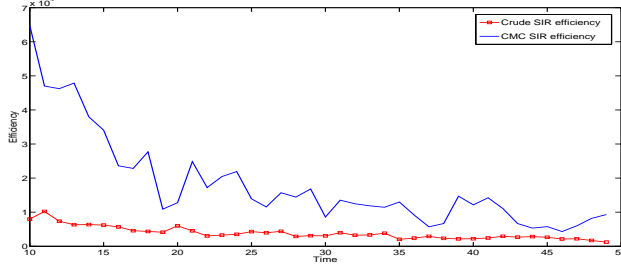
$\mathbf{H}_n = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$ ,  $\mathbf{R}_n = \begin{pmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{pmatrix}$  and  $\mathbf{G}_n(r) = \mathbf{I}$ ,  $T = 2s$ ,  $\sigma_v = 3m^2/sec^3$  and  $\sigma_x = \sigma_y = 10m$ . We track a maneuvering target described by its position and velocity in the Cartesian coordinates,  $\mathbf{x}_n = [p_x, \dot{p}_x, p_y, \dot{p}_y]^T$ . Mode  $r_n$  represents the behavior of the target: straight, left turn and right turn. Remember from 3.1.2 that the computation of  $\hat{\Theta}_n$  involves an extra computational cost compared to that of  $\tilde{\Theta}_n$ . So we compute the efficiency over  $P = 200$  simulations defined as [30]

$$\text{Eff}(n) = \frac{1}{\text{MSE}(n)\mathbb{E}(C(n))}, \quad (61)$$

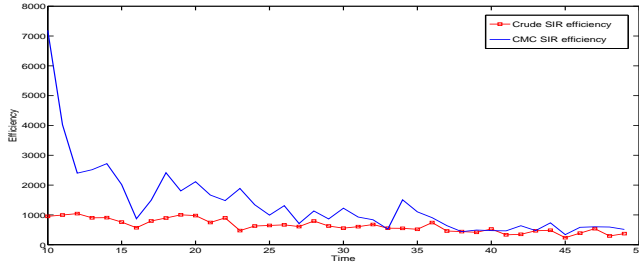
where  $C(n)$  is the CPU time to compute the estimator, and we discuss the performances of  $\hat{\Theta}_n$  and  $\tilde{\Theta}_n$  in function of the model parameters. Both estimators are computed with  $N = 1000$  particles.

We first set  $\omega_1 = 0 \text{ rad.s}^{-1}$ ,  $\omega_2 = 3\pi/180 \text{ rad.s}^{-1}$  and  $\omega_3 = -3\pi/180 \text{ rad.s}^{-1}$ . The Markovian transition probability is  $p(r_n|r_{n-1}) = 0.4$  if  $r_n = r_{n-1}$  and  $p(r_n|r_{n-1}) = 0.3$  otherwise. In Figure 6, we display the (averaged) efficiency of both estimators over time. The efficiency of  $\tilde{\Theta}_n$  is greater than that of  $\hat{\Theta}_n$ , so the Bayesian CMC estimator for linear JMSS is of practical interest. Note that the dependency of the model in  $\{r_n\}$  is weak since  $w_r$  is small and the Markovian transition probabilities are close. So distributions  $\{p(r_n|\mathbf{r}_{0:n-1}^i, \mathbf{y}_{0:n})\}_{i=1}^N$  tend to be uniform, and remember that  $\tilde{\Theta}_n$  computes directly the expectations according to these distributions while  $\hat{\Theta}_n$  uses samples  $\mathbf{r}_n^{1:N}$  according to them. This is why the gap between both estimators gets larger when distributions  $\{p(r_n|\mathbf{r}_{0:n-1}^i, \mathbf{y}_{0:n})\}_{i=1}^N$  become almost uniform.

Next we increase the dependency of the model in  $\{r_n\}$  by setting  $\omega_1 = 0 \text{ rad.s}^{-1}$ ,  $\omega_2 = 8\pi/180 \text{ rad.s}^{-1}$  and  $\omega_3 = -8\pi/180 \text{ rad.s}^{-1}$ . We also set  $p(r_k|r_{k-1}) = 0.6$  if  $r_k = r_{k-1}$  and  $p(r_k|r_{k-1}) = 0.2$  otherwise. In Figure 7, we display the averaged efficiency of both estimators for this new set of parameters. Indeed, the gap between both estimators is reduced but  $\tilde{\Theta}_n$  still outperforms  $\hat{\Theta}_n$ .



**Fig. 6** Efficiency - Linear JMSS Model - Close Markovian transition probabilities -  $\phi(\mathbf{x}_n) = \mathbf{x}_n$ . Due to the weak dependency of the model in  $\mathbf{r}_n$ , it is dangerous to sample new particles (Crude SIR) before computing the estimator.



**Fig. 7** Efficiency - Linear JMSS Model - Dispersed Markovian transition probabilities -  $\phi(\mathbf{x}_n) = \mathbf{x}_n$ . Contrary to the previous simulation, distributions  $\{p(r_n | \mathbf{r}_{0:n-1}^i, \mathbf{y}_{0:n})\}_{i=1}^N$  are dispersed so the gap between the crude SIR estimator and the CMC SIR one shrinks.

#### 4 Bayesian CMC algorithms for Multi-Target filtering

In this final section we apply CMC to multi-target filtering. Some adaptations are necessary, because in the multi-target context we do not necessary deal with classical pdf. However, the discussion in section 1.3 still holds, as we shall see. Let us begin with a brief review of multi-object filtering.

##### 4.1 A brief review of Random Finite Sets (RFS) based multi-target filtering

Multi-object filtering extends the previous problem in the sense that we now look for estimating an unknown number of targets from a set of observations which are either due to detected targets or are false alarms measurements. Classical solutions such as the Joint Probabilist Data association filter [31] or the Multiple Hypothesis Tracker [32] include a matching mechanism between targets and observations. Alternate solutions are based on RFS, which are sets of random variables with random and time-varying cardinal (see e.g. [33]). The interest of RFS based techniques over classical solutions is that they no longer require such a matching mechanism. The RFS formulation was first used to

derive the multi-object Bayesian filter, which generalizes the classical single object one [3]. This multi-object Bayesian filter involves the computation of set integrals of multi-object densities, i.e. of positive functions  $f(X)$  of a given RFS  $X$ , and cannot be computed in practice (SMC approximations can however be of interest when the number of targets is small [34]). Later on, Mahler proposed to propagate a first order moment of the multi-object density, the so-called PHD or intensity [3]. Let  $|X \cap S|$  be the number of objects in RFS  $X$  which belong to region  $S$ ; then the PHD  $v(\mathbf{x})$  is defined as the spatial density of the expected number of targets, i.e.

$$\int_{S \subset \mathbb{R}^p} v(\mathbf{x}) d\mathbf{x} = \mathbb{E}(|X \cap S|). \quad (62)$$

Its interest in multi-object filtering is twofold; first,  $\int v(\mathbf{x}) d\mathbf{x}$  is an estimate of the number of targets; in addition, extracting the states consists in looking for regions where the PHD is high, and so local maxima of  $v$  are required. Let now  $v_n(\mathbf{x})$  be the a posteriori PHD, i.e. the first order moment  $v_n(\mathbf{x})$  of the multi-object density at time  $n$ , given the set of past measurements  $Z_{0:n} = \{Z_1, \dots, Z_n\}$ , where  $Z_k$  is the set of measurements available at time  $k$ . The PHD filter is a set of equations which enables to propagate  $v_n$  and which has the advantage to make use of classical integrals only. If we assume that the cardinality distributions of the number of targets and of false alarm measurements are Poisson, and that each target evolves and generates observations independently of one another, then PHD  $v_n$  is propagated as follows (we assume for simplicity that there is no spawning) [3] [33]:

$$\begin{aligned} v_{n|n-1}(\mathbf{x}_n) &= \int p_{s,n}(\mathbf{x}_{n-1}) f_{n|n-1}(\mathbf{x}_n | \mathbf{x}_{n-1}) v_{n-1}(\mathbf{x}_{n-1}) d\mathbf{x}_{n-1} + \gamma_n(\mathbf{x}_n), \\ v_n(\mathbf{x}_n) &= [1 - p_{d,n}(\mathbf{x}_n)] v_{n|n-1}(\mathbf{x}_n) \\ &\quad + \sum_{\mathbf{z} \in Z_n} \frac{p_{d,n}(\mathbf{x}_n) g_n(\mathbf{z} | \mathbf{x}_n) v_{n|n-1}(\mathbf{x}_n)}{\kappa_n(\mathbf{z}) + \int p_{d,n}(\mathbf{x}_n) g_n(\mathbf{z} | \mathbf{x}_n) v_{n|n-1}(\mathbf{x}_n) d\mathbf{x}_n}, \end{aligned} \quad (63) \quad (64)$$

where  $p_{s,n}(\cdot)$  (resp.  $p_{d,n}(\cdot)$ ) is the probability of survival (resp. of detection) at time  $n$  which can depend on state  $\mathbf{x}_{n-1}$  (resp. on  $\mathbf{x}_n$ ); and  $\kappa_n(\cdot)$  (resp.  $\gamma_n(\cdot)$ ) is the intensity of the false alarms measurements (resp. of the birth targets) at time  $n$ .

#### 4.2 Deriving the Bayesian CMC PHD estimator

Let us now turn back to the derivation of a Bayesian CMC PHD estimator. First, the problem we address is to compute the moment  $\Theta_n = \int f(\mathbf{x}_n) v_n(\mathbf{x}_n) d\mathbf{x}_n$  (typically, we shall take either  $f(\mathbf{x}_n) = 1$  or  $f(\mathbf{x}_n) = \mathbf{1}_S(\mathbf{x}_n)$ , where  $S$  is some region of interest). From now on we assume that  $p_{d,n}$  does not depend on  $\mathbf{x}_n$ . Plugging (63) in (64), the PHD at time  $n$  can be written as

$$v_n(\mathbf{x}_n) = \sum_{i=1}^4 v_n^i(\mathbf{x}_n), \quad (65)$$

where

$$v_n^1(\mathbf{x}_n) = [1 - p_{d,n}] \int p_{s,n}(\mathbf{x}_{n-1}) f_{n|n-1}(\mathbf{x}_n | \mathbf{x}_{n-1}) v_{n-1}(\mathbf{x}_{n-1}) d\mathbf{x}_{n-1}, \quad (66)$$

$$v_n^2(\mathbf{x}_n) = [1 - p_{d,n}] \gamma_n(\mathbf{x}_n), \quad (67)$$

$$v_n^3(\mathbf{x}_n) = \sum_{\mathbf{z} \in Z_n} \frac{p_{d,n} g_n(\mathbf{z} | \mathbf{x}_n) \int p_{s,n}(\mathbf{x}_{n-1}) f_{n|n-1}(\mathbf{x}_n | \mathbf{x}_{n-1}) v_{n-1}(\mathbf{x}_{n-1}) d\mathbf{x}_{n-1}}{B_n(\mathbf{z})}, \quad (68)$$

$$v_n^4(\mathbf{x}_n) = \sum_{\mathbf{z} \in Z_n} \frac{p_{d,n} g_n(\mathbf{z} | \mathbf{x}_n) \gamma_n(\mathbf{x}_n)}{B_n(\mathbf{z})}, \quad (69)$$

and where

$$B_n(\mathbf{z}) = \kappa_n(\mathbf{z}) + B_n^1(\mathbf{z}) + B_n^2(\mathbf{z}), \quad (70)$$

$$\begin{aligned} B_n^1(\mathbf{z}) &= \int p_{d,n} g_n(\mathbf{z} | \mathbf{x}_n) \int p_{s,n}(\mathbf{x}_{n-1}) f_{n|n-1}(\mathbf{x}_n | \mathbf{x}_{n-1}) v_{n-1}(\mathbf{x}_{n-1}) d\mathbf{x}_{n-1} d\mathbf{x}_n \\ &= p_{d,n} \int p_{s,n}(\mathbf{x}_{n-1}) p(\mathbf{z} | \mathbf{x}_{n-1}) v_{n-1}(\mathbf{x}_{n-1}) d\mathbf{x}_{n-1}, \end{aligned} \quad (71)$$

$$B_n^2(\mathbf{z}) = \int p_{d,n} g_n(\mathbf{z} | \mathbf{x}_n) \gamma_n(\mathbf{x}_n) d\mathbf{x}_n. \quad (72)$$

Term  $v_n^1$  (resp.  $v_n^2$ ) is due to non-detected persistent (resp. birth) targets, while  $v_n^3$  (resp.  $v_n^4$ ) is due to detected persistent (resp. birth) targets.

From (65) we see that

$$\Theta_n = \sum_{i=1}^4 \underbrace{\int f(\mathbf{x}_n) v_n^i(\mathbf{x}_n) d\mathbf{x}_n}_{\Theta_n^i}, \quad (73)$$

so we now consider whether one can adapt the Bayesian CMC methodology of section 1.3 to any of the moments  $\Theta_n^i$ . First, note that  $v_n^2(\mathbf{x}_n)$  and  $v_n^4(\mathbf{x}_n)$  do not depend on  $v_{n-1}(\mathbf{x}_{n-1})$  so we use a crude MC procedure to compute  $\Theta_n^2$  and  $\Theta_n^4$ . Let  $\hat{v}_{n-1} = \sum_{i=1}^{L_{n-1}} w_{n-1}^i \delta_{\mathbf{x}_{n-1}^i}$  and  $\hat{\gamma}_n = \sum_{i=1}^{L_{\gamma_n}} w_{\gamma_n}^i \delta_{\mathbf{x}_{\gamma_n}^i}$  be MC approximations of  $v_{n-1}(\mathbf{x}_{n-1})$  and of  $\gamma_n(\mathbf{x}_n)$ , respectively. Then  $\hat{\Theta}_n^2 = \sum_{i=1}^{L_{\gamma_n}} w_n^{2,i} f(\mathbf{x}_{\gamma_n}^i)$ ,  $\hat{\Theta}_n^4 = \sum_{i=1}^{L_{\gamma_n}} [\sum_{\mathbf{z} \in Z_n} w_n^{4,i}(\mathbf{z})] f(\mathbf{x}_{\gamma_n}^i)$ , where  $w_n^{2,i} = w_{\gamma_n}^i [1 - p_{d,n}]$ ,  $w_n^{4,i}(\mathbf{z}) = w_{\gamma_n}^i \frac{p_{d,n} g_n(\mathbf{z} | \mathbf{x}_{\gamma_n}^i)}{B_n(\mathbf{z})}$  and

$$\tilde{B}_n(\mathbf{z}) = \kappa_n(\mathbf{z}) + p_{d,n} \sum_{i=1}^{L_{\gamma_n}} w_{\gamma_n}^i g_n(\mathbf{x}_{\gamma_n}^i | \mathbf{z}) + p_{d,n} \sum_{i=1}^{L_{n-1}} w_{n-1}^i p_{s,n}(\mathbf{x}_{n-1}^i) p(\mathbf{z} | \mathbf{x}_{n-1}^i). \quad (74)$$

By contrast, the computation of  $v_n^1(\mathbf{x}_n)$  and of  $v_n^3(\mathbf{x}_n)$  depends on  $v_{n-1}(\mathbf{x}_{n-1})$ . This suggests adapting the common methodology described in section 1, even though the PHD is not a pdf (it is a positive function, but remember from (62) that its integral is not equal to 1), and that weights  $\{w_{n-1}^i\}_{i=1}^{L_{n-1}}$  may depend

on variables different from  $\mathbf{x}_{n-1}^{1:L_{n-1}}$ , but which are known at time  $n-1$ . These differences do not impact the discussion of section 1.3 which can be used in this context. Indeed, we have  $f_n(\mathbf{x}_n)g_n(\mathbf{z}|\mathbf{x}_n) = p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{z})p(\mathbf{z}|\mathbf{x}_{n-1})$ , so  $\Theta_n^1$  and  $\Theta_n^3$  can be rewritten as

$$\Theta_n^1 = [1 - p_{d,n}] \int \mathbb{E}(f(\mathbf{x}_n)|\mathbf{x}_{n-1}) \times [p_{s,n}(\mathbf{x}_{n-1})v_{n-1}(\mathbf{x}_{n-1})] d\mathbf{x}_{n-1}, \quad (75)$$

$$\Theta_n^3 = \sum_{\mathbf{z} \in Z_n} \int \mathbb{E}(f(\mathbf{x}_n)|\mathbf{x}_{n-1}, \mathbf{z}) \left[ \frac{p_{d,n}p_{s,n}(\mathbf{x}_{n-1})p(\mathbf{z}|\mathbf{x}_{n-1})v_{n-1}(\mathbf{x}_{n-1})}{B_n(\mathbf{z})} \right] d\mathbf{x}_{n-1}. \quad (76)$$

Let us start with the computation of  $\Theta_n^1$  in (75). Even if it is not a pdf, the factor  $p_{s,n}(\mathbf{x}_{n-1})v_{n-1}(\mathbf{x}_{n-1})$  within brackets plays the role of  $p(x_1)$  in (7), and can be approximated by  $\sum_{i=1}^{L_{n-1}} w_n^{1,i} \delta_{\mathbf{x}_{n-1}^i}$  where  $w_n^{1,i} = [1 - p_{d,n}] p_{s,n}(\mathbf{x}_{n-1}^i) w_{n-1}^i$ . So the crude MC and Bayesian CMC estimators of  $\Theta_n^1$  are respectively

$$\hat{\Theta}_{1,n} = \sum_{i=1}^{L_{n-1}} w_n^{1,i} f(\mathbf{x}_n^i), \quad (77)$$

$$\tilde{\Theta}_{1,n} = \sum_{i=1}^{L_{n-1}} w_n^{1,i} \mathbb{E}(f(\mathbf{x}_n)|\mathbf{x}_{n-1}^i), \quad (78)$$

in which  $\mathbf{x}_n^i \sim f_n(\mathbf{x}_n|\mathbf{x}_{n-1}^i)$ . Let us next address  $\Theta_n^3$  in (76). For each measurement  $\mathbf{z} \in Z_n$ , the factor  $\frac{p_{d,n}p_{s,n}(\mathbf{x}_{n-1})p(\mathbf{z}|\mathbf{x}_{n-1})v_{n-1}(\mathbf{x}_{n-1})}{B_n(\mathbf{z})}$  within brackets plays the role of  $p(x_1)$  in (7), and can be approximated by  $\sum_{i=1}^{L_{n-1}} w_n^{3,i}(\mathbf{z}) \delta_{\mathbf{x}_{n-1}^i}$  where  $w_n^{3,i}(\mathbf{z}) = p_{d,n}p_{s,n}(\mathbf{x}_{n-1}^i)p(\mathbf{z}|\mathbf{x}_{n-1}^i)w_{n-1}^i/\tilde{B}_n(\mathbf{z})$ . So the crude MC and Bayesian CMC estimators of  $\Theta_n^3$  are respectively

$$\hat{\Theta}_n^3 = \sum_{\mathbf{z}} \sum_{i=1}^{L_{n-1}} w_n^{3,i}(\mathbf{z}) f(\mathbf{x}_n^{\mathbf{z},i}), \quad (79)$$

$$\tilde{\Theta}_n^3 = \sum_{\mathbf{z} \in Z_n} \sum_{i=1}^{L_{n-1}} w_n^{3,i}(\mathbf{z}) \mathbb{E}(f(\mathbf{x}_n)|\mathbf{x}_{n-1}^i, \mathbf{z}), \quad (80)$$

in which  $\mathbf{x}_n^{\mathbf{z},i} \sim p(\mathbf{x}_n|\mathbf{x}_{n-1}^i, \mathbf{z})$ .

In summary, the crude MC PHD estimator  $\hat{\Theta}_n$  of  $\Theta_n$  is the sum of four crude MC estimators:  $\hat{\Theta}_n = \sum_{i=1}^4 \hat{\Theta}_n^i$ , while our Bayesian CMC PHD estimator  $\tilde{\Theta}_n$  is a sum of two crude MC and two Bayesian CMC estimators:  $\tilde{\Theta}_n = \tilde{\Theta}_n^1 + \tilde{\Theta}_n^2 + \tilde{\Theta}_n^3 + \tilde{\Theta}_n^4$ . Since  $\tilde{\Theta}_n^1$  and  $\tilde{\Theta}_n^3$  are computed from the same MC approximation of  $v_{n-1}(\mathbf{x}_{n-1})$ ,  $\tilde{\Theta}_n = \mathbb{E}(\hat{\Theta}_n|\{\mathbf{x}_{n-1}^i\}_{i=1}^{L_{n-1}}, \{\mathbf{x}_{\gamma_n}^i\}_{i=1}^{L_{\gamma_n}}, Z_n)$ , so section 1 enables to conclude that  $\tilde{\Theta}_n$  indeed outperforms  $\hat{\Theta}_n$ .

*Remark 5* The computation of  $\hat{\Theta}_n^1 + \hat{\Theta}_n^3$  involves to sample  $L_{n-1}(|Z_n|+1)$  particles where  $|Z_n|$  is the cardinal of  $Z_n$ . It is possible to compute an approximation



with  $L_{n-1}$  particles by sampling  $\mathbf{x}_n^i \sim q^i(x)$  with  $q^i(x) \propto w_n^{1,i} f_{n|n-1}(\mathbf{x}_n | \mathbf{x}_{n-1}^i) + \sum_{\mathbf{z}} w_n^{3,i}(\mathbf{z}) p(\mathbf{x}_n | \mathbf{x}_{n-1}^i, \mathbf{z})$  for all  $i$ ,  $1 \leq i \leq L_{n-1}$ , and by taking

$$\hat{\Theta}_n^1 + \hat{\Theta}_n^3 = \sum_{i=1}^{L_{n-1}} \left[ w_n^{1,i} + \sum_{\mathbf{z}} w_n^{3,i}(\mathbf{z}) \right] f(\mathbf{x}_n^i). \quad (81)$$

*Remark 6* Depending on the form of  $\gamma_n(\mathbf{x}_n)$  and  $g_n(\mathbf{z} | \mathbf{x}_n)$ ,  $\int g_n(\mathbf{z} | \mathbf{x}_n) \gamma_n(\mathbf{x}_n) d\mathbf{x}_n$  may be directly computable, so  $B_n^2(\mathbf{z})$ ,  $\Theta_{2,n}$  and  $\Theta_{4,n}$  may be computable too. In this case one can replace  $\tilde{B}_n(\mathbf{z})$  in (74) by  $\kappa_n(\mathbf{z}) + B_n^2(\mathbf{z}) + p_{d,n} \sum_{i=1}^{L_{n-1}} w_{n-1}^i p_{s,n}(\mathbf{x}_{n-1}^i) p(\mathbf{z} | \mathbf{x}_{n-1}^i)$ .

#### 4.3 Computing the CMC PHD filter $\tilde{\Theta}_n$ in practice

In the multi-target filter problem, we look for computing an estimator of the number of targets and of multi-target states. From (62), an estimator of the number of targets is given by

$$\tilde{N}_n = \sum_{i=1}^{L_{n-1}} w_n^{1,i} + \sum_{\mathbf{z} \in Z_n} \sum_{i=1}^{L_{n-1}} w_n^{3,i}(\mathbf{z}) + \sum_{i=1}^{L_{\gamma_n}} w_n^{2,i} + \sum_{\mathbf{z} \in Z_n} \sum_{i=1}^{L_{\gamma_n}} w_n^{4,i}(\mathbf{z}). \quad (82)$$

The procedure to extract persistent targets consists in looking for local maxima of  $\sum_{i=1}^{L_{n-1}} w_n^{1,i} p(\mathbf{x}_n | \mathbf{x}_{n-1}^i) + \sum_{\mathbf{z} \in Z_n} \sum_{i=1}^{L_{n-1}} w_n^{3,i}(\mathbf{z}) p(\mathbf{x}_n | \mathbf{x}_{n-1}^i, \mathbf{z})$ . For birth targets, this procedure cannot be used if the PHD due to birth targets was computed via an MC approximation. One can use clustering techniques [34], or the procedure described in [35], which consists in looking for measurements  $\mathbf{z}$  such that  $\sum_{i=1}^{L_{\gamma_n}} w_n^{4,i}(\mathbf{z})$  is above a given threshold (typically 0.5); then an estimator of the state associated to  $\mathbf{z}$  is given by  $\sum_{i=1}^{L_{\gamma_n}} w_n^{4,i}(\mathbf{z}) \mathbf{x}_{\gamma_n}^i$ . However, birth targets become persistent targets at the next time step; so their extraction becomes easy at the next iteration since an SMC extraction procedure can be avoided.

*Remark 7* One can also adapt the procedure described above [35] to the extraction of persistent target states, i.e. looking for measurements  $\mathbf{z}$  such that  $\sum_{i=1}^{L_{n-1}} w_n^{3,i}(\mathbf{z})$  is above a given threshold, and estimating the associated state by  $\sum_{i=1}^{L_{n-1}} w_n^{3,i}(\mathbf{z}) \int \mathbf{x}_n p(\mathbf{x}_n | \mathbf{x}_{n-1}^i, \mathbf{z}) d\mathbf{x}_n$ . The advantage of this procedure is that we just need to compute  $\int \mathbf{x}_n p(\mathbf{x}_n | \mathbf{x}_{n-1}^i, \mathbf{z}) d\mathbf{x}_n$  for such measurements.

Let us now detail some applications of the CMC-PHD filter.

##### 4.3.1 Gaussian and linear models with Gaussian Mixture (GM) birth intensity: an alternative to the GM implementation of the PHD filter

We first assume that  $f_{n|n-1}(\mathbf{x}_n | \mathbf{x}_{n-1}) = \mathcal{N}(\mathbf{x}_n; \mathbf{F}_n \mathbf{x}_{n-1}; \mathbf{Q}_n)$ ,  $g_n(\mathbf{z} | \mathbf{x}_n) = \mathcal{N}(\mathbf{z}; \mathbf{H}_n \mathbf{x}_n; \mathbf{R}_n)$ , and that  $\gamma_n$  is a GM, i.e. that  $\gamma_n(\mathbf{x}_n) = \sum_{i=1}^{N_{\gamma_n}} w_{\gamma_n}^i \mathcal{N}(\mathbf{x}_n; \mathbf{m}_{\gamma_n}^i$ ;

$\mathbf{P}_{\gamma_n}^i$ ). For such models a GM implementation has been proposed [36], which consists in propagating a GM approximation of PHD  $v_n$  via (63)-(64). The mixture grows exponentially due to the summation on the set of measurements in (64), so pruning and merging approximations are necessary. In addition, this implementation requires that  $p_{d,n}$  and  $p_{s,n}$  are constant (or possibly GM [36]). In our algorithm we do not need to make any assumption about  $p_s(\mathbf{x}_{n-1})$ . For this model  $B_n^2(\mathbf{z})$  is directly computable, and the Bayesian CMC procedure for estimating the number of targets and extracting the states is valid since  $p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{z})$  and  $p(\mathbf{z}|\mathbf{x}_{n-1})$  are computable (see (20)-(21) and (25)-(26)). Finally, in the case where  $p_s(\mathbf{x}_{n-1})$  is constant, we have at our disposal three implementations of the PHD filter: the GM [36], the SMC [34] and our Bayesian CMC implementations which will be compared in section 4.4 below.

#### 4.3.2 Gaussian and linear models with ordinary birth intensity

If  $\gamma_n$  is not a GM the GM implementation cannot be used any longer. However, our method remains valid if we compute  $\Theta_{2,n}$ ,  $\Theta_{4,n}$  and  $B_n^2(\mathbf{z})$  via an MC approximation. By contrast to the pure SMC technique, our Bayesian CMC implementation enables to keep the GM structure for persistent targets.

#### 4.3.3 Non linear models

In a non-linear model the GM implementation cannot be used any longer. The extended (resp. unscented) Kalman PHD filter [36] approximates the PHD by a GM, the parameters of which are propagated by an EKF (resp. UKF). By contrast, we propose to adapt our Bayesian CMC implementation, by approximating  $p(\mathbf{x}_n|\mathbf{x}_{n-1}, \mathbf{z})$  and  $p(\mathbf{z}|\mathbf{x}_{n-1})$  at time  $n$  by techniques described in §2.2. The main difference is that we start from a discrete approximation of the PHD at time  $n-1$ , and compute an estimate of the states without using clustering techniques of the MC implementation. This way we get an approximation of the PHD which does not rely on a numerical approximation at time  $n-1$  and which enables to extract the states easily. In addition, by contrast to the extended and unscented implementations of the PHD filter, numerical approximations are not propagated over time since they are only used locally for the extraction of states.

### 4.4 Simulations

We now compare our Bayesian CMC PHD estimator to alternative implementations of the PHD filter. The MSE criterion used previously is not appropriate in the multi-target context: since the number of targets evolves, a performances criterion should take into account an estimator of the number of targets and an estimator of their states. So in this section we will use the optimal subpattern assignment (OSPA) distance [37], which is a classical tool for comparing multi-target filtering algorithms. Let  $X = \{x_1, \dots, x_m\}$  and  $Y = \{y_1, \dots, y_n\}$

be two finite sets, which respectively represent the estimated and true sets of targets. For  $1 \leq p < +\infty$  and  $c > 0$ , let  $d^{(c)}(x, y) = \min(c, \|x - y\|)$  ( $\|\cdot\|$  is the euclidean norm) and let  $\Pi_n$  be the set of permutations on  $\{1, 2, \dots, n\}$ . The OSPA metric is defined by :

$$\bar{d}_p^c(X, Y) \triangleq \left( \frac{1}{n} \left( \min_{\pi \in \Pi_n} \sum_{i=1}^m d^{(c)}(x_i, y_{\pi(i)})^p + c^p(n-m) \right) \right)^{\frac{1}{p}} \quad (83)$$

if  $m \leq n$ , and  $\bar{d}_p^c(X, Y) \triangleq \bar{d}_p^c(Y, X)$  if  $m > n$ . The term  $\min_{\pi \in \Pi_n} \sum_{i=1}^m d^{(c)}(x_i, y_{\pi(i)})^p$  represents the localization error, while the second term represents the cardinality error.

We focus on the linear and Gaussian model in which the GM-PHD is used as a benchmark solution and enables to appreciate the performance of our Bayesian CMC-PHD filter. So we compare the GM-PHD, the SMC-PHD and our Bayesian CMC-PHD filters. We track the position and velocity of the targets so  $\mathbf{x}_n = [p_x, \dot{p}_x, p_y, \dot{p}_y]^T_n$ . Let also  $f_{n|n-1}(\mathbf{x}_n | \mathbf{x}_{n-1}) = \mathcal{N}(\mathbf{x}_n; \mathbf{F}_n \mathbf{x}_{n-1}, \mathbf{Q}_n)$

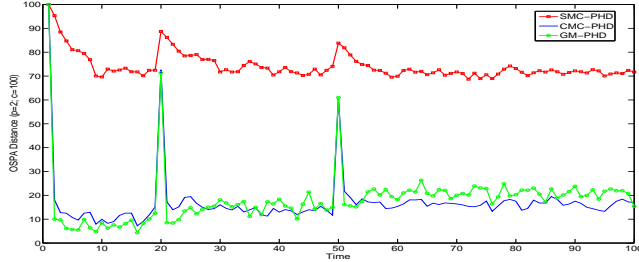
and  $g_n(\mathbf{z}_n | \mathbf{x}_n) = \mathcal{N}(\mathbf{z}_n; \mathbf{H}_n \mathbf{x}_n, \mathbf{R}_n)$ , where  $\mathbf{F}_n = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}$ , and the other pa-

rameters ( $\mathbf{H}_n$ ,  $\mathbf{Q}_n$  and  $\mathbf{R}_n$ ) are identical to those of §3.3.

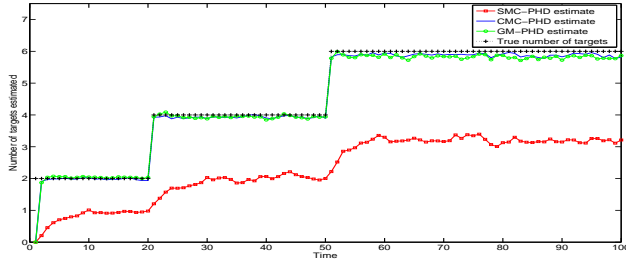
We compare the SMC-PHD and our Bayesian CMC filters in the case where both algorithms use the transition pdf  $f_{n|n-1}(\mathbf{x}_n | \mathbf{x}_{n-1})$  (remember that in our approach, we need to propagate a discrete approximation of the PHD, even if it not used for computing an estimator of the number of targets). We take  $T = 2s$ ,  $\sigma_v = 3m^2/sec^3$  but  $\sigma_x = \sigma_y = 0.3m$ , which means that the likelihood  $g_n(\mathbf{z}_n | \mathbf{x}_n)$  is sharp; since the transition pdf does not take into account available measurements, it is difficult to guide particles into promising regions, so this experimental scenario is challenging for the SMC-PHD implementation. Particles are initialized around the measurements [35]. In both algorithms we use  $N_b = 20$  particles per newborn target and  $N = 200$  particles per persistent target. The probability of detection is  $p_{d,k} = 0.95$  and that of survival  $p_{s,k} = 0.98$ , for all  $k$ ,  $1 \leq k \leq 100$ , and we generate 10 false alarm measurements (in mean). We consider a scenario with 6 targets which appear either at  $k = 0$ ,  $k = 20$  or  $k = 50$ . We also test the GM implementation in which  $T_p = 10^{-5}$  for the pruning threshold,  $T_m = 4m$  for the merging threshold and we keep at most  $N_{\max} = 100$  Gaussians (see §4.3.1).

The OSPA distance and estimated number of targets are displayed in Figures 8 and 9. The Bayesian CMC approach outperforms the SMC one and copes with the issue of guiding particles in promising regions. Even if we use the transition density for getting a discrete approximation of  $v_{n-1}$ , the Bayesian CMC approach provides a correct estimate of the number of targets, by contrast to the SMC one in which the new set  $\{\mathbf{x}_n^i, w_n^i\}_{i=1}^{L_n}$  is used to deduce a discrete approximation of  $v_n$ , then an estimate of the number of targets. The Bayesian CMC PHD estimator also outperforms the GM one in terms of

OSPA distance. Finally the number of targets is well estimated both by the GM and Bayesian CMC implementations, but the Bayesian CMC estimator is more accurate, see Figure 10.



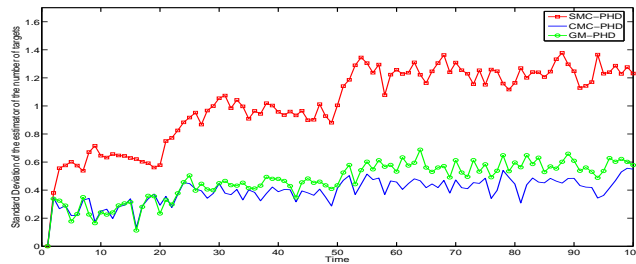
**Fig. 8** OSPA distance for linear and Gaussian scenario - The GM and CMC implementations widely outperform the classical SMC one because of the choice of the likelihood function  $g_n(\mathbf{z}|\mathbf{x})$ .



**Fig. 9** Estimator of the number of targets for linear and Gaussian scenario.

## 5 Conclusion

In this paper we adapted CMC to single- and multi-object Bayesian filtering. In this framework, the recursive nature of SMC algorithms provides a conditioning variable at each time instant, but i.i.d. samples from this conditioning variable are unavailable. Our variance reduction method can be seen as a temporal, rather than spatial, RB-PF procedure; a Bayesian CMC estimator is ensured to outperform the associated crude MC one whatever the number of particles. We next showed that a CMC estimator can indeed be computed, or approximated, in a variety of Markovian stochastic models, including semi-linear HMC or JMSS, either at the same cost or at a reasonable extra computational cost. Finally we adapted Bayesian CMC to multi-target



**Fig. 10** Standard Deviation of the estimator of the number of targets for linear and Gaussian scenario - The CMC estimator is slightly more reliable than the GM one when time increases.

filtering, and showed that our CMC PHD filter has interesting practical features as compared to alternate (SMC or GM) implementations of the PHD filter. Our analysis was validated via simulations.

## 6 Acknowledgements

The authors would like to thank the French MOD DGA/MRIS for financial support of the Ph.D. of Y. Petetin.

## References

1. A. Doucet, N. de Freitas, N. Gordon, Sequential Monte Carlo Methods in Practice, Statistics for Engineering and Information Science, Springer Verlag, New York, 2001.
2. M. S. Arulampalam, S. Maskell, N. Gordon, T. Clapp, A tutorial on particle filters for online nonlinear / non-Gaussian Bayesian tracking, IEEE Tr. Signal Processing 50 (2) (2002) 174–188.
3. R. Mahler, Multitarget Bayes filtering via first-order multitarget moments, IEEE Transactions on Aerospace and Electronic Systems 39 (4) (2003) 1152–1178.
4. S. Asmussen, P. W. Glynn, *Stochastic Simulation: Algorithms and Analysis*, Springer-Verlag, New York, NY, 2007.
5. R. Chen, J. S. Liu, Mixture Kalman filters, J. R. Statist. Soc. B 62 (2000) 493–508.
6. A. Doucet, S. J. Godsill, C. Andrieu, On sequential Monte Carlo sampling methods for Bayesian filtering, Statistics and Computing 10 (2000) 197–208.
7. A. Doucet, N. J. Gordon, V. Krishnamurthy, Particle filters for state estimation of jump Markov linear systems, IEEE Transactions on Signal Processing 49 (3) (2001) 613–24.
8. T. Schön, F. Gustafsson, P.-J. Nordlund, Marginalized particle filters for mixed linear nonlinear state-space models, IEEE Trans. on Signal Processing 53 (2005) 2279–2289.
9. J. Geweke, Bayesian inference in econometric models using Monte Carlo integration, Econometrica 57 (6) (1989) 1317–1339.
10. N. Chopin, Central limit theorem for sequential Monte Carlo methods and its application to Bayesian inference, The Annals of Statistics 32 (6) (2004) 2385–2411.
11. F. Lindsten, T. Schön, J. Olsson, An explicit variance reduction expression for the Rao-Blackwellized particle filter, in: 18th World Congress of the Int. Federation of Automatic Control (IFAC), 2011.
12. D. B. Rubin, Using the SIR algorithm to simulate posterior distributions, in: M. H. Bernardo, K. M. Degroot, D. V. Lindley, A. F. M. Smith (Eds.), Bayesian Statistics III, Oxford University Press, Oxford, 1988.

13. A. E. Gelfand, A. F. M. Smith, Sampling based approaches to calculating marginal densities, *Journal of the American Statistical Association* 85 (410) (1990) 398–409.
14. A. F. M. Smith, A. E. Gelfand, Bayesian statistics without tears : a sampling-resampling perspective., *The American Statistician* 46 (2) (1992) 84–87.
15. V. Zaritskii, V. Svetnik, L. Shimelevich, Monte carlo techniques in problems of optimal data processing, *Automation and remote control* (1975) 95–103.
16. A. Kong, J. S. Liu, W. H. Wong, Sequential imputations and Bayesian missing data problems, *Journal of the American Statistical Association* 89 (425) (1994) 278–88.
17. J. S. Liu, R. Chen, Blind deconvolution via sequential imputation, *Journal of the American Statistical Association* 90 (430) (1995) 567–76.
18. S. Saha, P. K. Manda, Y. Boers, H. Driessen, A. Bagchi, Gaussian proposal density using moment matching in SMC methods., *Statistics and Computing* 19-2 (2009) 203–208.
19. S. Julier, J. Uhlmann, Unscented filtering and nonlinear estimation, in: *Proceedings of the IEEE*, Vol. 92, 2004, pp. 401–422.
20. M. K. Pitt, N. Shephard, Filtering via simulation : Auxiliary particle filters, *Journal of the American Statistical Association* 94 (446) (1999) 590–99.
21. P. Fearnhead, Computational methods for complex stochastic systems: A review of some alternatives to MCMC, *Statistics and Computing* 18 (2) (2008) 151–71.
22. A. M. Johansen, A. Doucet, A note on the auxiliary particle filter, *Statistics and Probability Letters* 78 (12) (2008) 1498–1504.
23. O. Cappé, É. Moulines, T. Rydén, *Inference in Hidden Markov Models*, Springer-Verlag, 2005.
24. N. J. Gordon, D. J. Salmond, A. F. M. Smith, Novel approach to nonlinear/ non-Gaussian Bayesian state estimation, *IEE Proceedings-F* 140 (2) (1993) 107–113.
25. J. K. Tugnait, Adaptive estimation and identification for discrete systems with Markov jump parameters, *IEEE Transactions on Automatic Control* 27 (5) (1982) 1054–65.
26. Y. Bar-Shalom, X. R. Li, T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*, John Wiley and sons, New-York, 2001.
27. C. Musso, N. oudjane, F. LeGland, Improving regularised particle filters, in: A. Doucet, N. de Freitas, N. Gordon (Eds.), *Sequential Monte Carlo Methods in Practice*, Statistics for Engineering and Information Science, Springer Verlag, New York, 2001.
28. S. McGinnity, G. W. Irwin, Multiple model bootstrap filter for maneuvering target tracking, *IEEE Transactions on Aerospace and Electronic Systems* 36 (3) (2000) 1006–1012.
29. C. Andrieu, M. Davy, A. Doucet, Efficient Particle Filtering For Jump Markov Systems, *IEEE trans. on Signal Processing* 51 (2002) 1762–1770.
30. P. L’Ecuyer, Efficiency improvement and variance reduction, in: *Winter Simulation Conference 1994*, 1994, pp. 122–132.
31. Y. Bar-Shalom, *Tracking and data association*, Academic Press Professional, Inc., San Diego, CA, 1987.
32. S. Blackman, R. Popoli, *Design and Analysis of Modern Tracking Systems*, Artech House, 2009.
33. R. Mahler, *Statistical Multisource Multitarget Information Fusion*, Artech House, 2007.
34. B.-N. Vo, S. Singh, A. Doucet, Sequential Monte Carlo methods for multi-target filtering with random finite sets, *IEEE Transactions on Aerospace and Electronic Systems* 41.
35. B. Ristic, D. Clark, B. Vo, Improved SMC implementation of the PHD filter, in: *Proceedings of the 13th International Conference on Information Fusion*, 2010.
36. B.-N. Vo, W. Ma, The Gaussian mixture probability hypothesis density filter, *IEEE Transactions on Signal Processing* 54 (2006) 4091–4104.
37. D. Schuhmacher, B. T. Vo, B. N. Vo, A consistent metric for performance evaluation of multi-object filters, *IEEE Transactions on Signal Processing* 56 (8) (2008) 3447–3457.